

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»
ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
«_____» _____ 2019 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОЇ РОБОТИ**

на тему

**ТРЕНАЖЕР З ТЕМИ «ЧИСЛЕННЯ ВИСЛОВЛЮВАНЬ» ДИСТАНЦІЙНОГО
НАВЧАЛЬНОГО КУРСУ «МАТЕМАТИЧНА ЛОГІКА» ТА РОЗРОБКА ЙОГО
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

зі спеціальності «Комп'ютерні науки»»

Виконавець роботи: студент групи КН-61 Москаленко Артем Олегович

_____ «___» _____ 2019 р.

Науковий керівник: к.ф.-м.н., доц., Черненко Оксана Олексіївна

_____ «___» _____ 2019 р.

Полтава-2019

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	7
ВСТУП	8
1 ПОСТАНОВКА ЗАДАЧ.....	10
1.1 Основні вимоги до змісту роботи.....	10
1.2 Постановка задач для реалізації програми-тренажера з теми «Числення висловлювань».....	10
2 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	12
2.1 Ефективність та недоліки ДН.....	12
2.2 Огляд сучасних систем ДН.....	13
2.3 Огляд існуючих розробок.....	15
2.4 Висновки за розділом 2.....	16
3 ТЕОРЕТИЧНА ЧАСТИНА.....	17
3.1 Алгоритмізація роботи тренажера.....	17
3.2 Теоретичні відомості с теми Числення висловлювань.....	17
3.3 Приклади та розв'язки завдань з теми.....	21
3.4 Алгоритм роботи тренажера.....	22
3.5 Розробка блок-схеми алгоритму роботи навчального тренажера	27
3.5 Обґрунтування вибору програмних засобів.....	28
3.6 Висновки за розділом 3.....	29
4 ПРАКТИЧНА ЧАСТИНА.....	31
4.1 Опис процесу програмної реалізації.....	31
4.2 Опис програми.....	31
4.3 Перевірка валідності.....	36
4.4 Необхідна користувачу інструкція програми	52
4.5 Висновки за розділом 4.....	54
ВИСНОВКИ.....	55
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	56

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

Умовне скорочення	Пояснення умовного скорочення
ДН	Дистанційне навчання
XML	Стандарт побудови мов розмітки
Java	Строго типізована об'єктно-орієнтована мова програмування

ВСТУП

Дистанційне навчання — сукупність сучасних технологій, що забезпечують доставку інформації в інтерактивному режимі за допомогою використання ІКТ (інформаційно-комунікаційних технологій) від тих, хто навчає (викладачів, визначних постатей у певних галузях науки, політиків), до тих, хто навчається (студентів чи слухачів). Застосовується під час підготовки як у середніх загальноосвітніх школах і ЗВО, так і в бізнес-школах. Основними принципами дистанційного навчання є інтерактивна взаємодія у процесі роботи, надання студентам можливості самостійного освоєння досліджуваного матеріалу, а також консультаційний супровід у процесі дослідницької діяльності. Дає змогу навчатися на відстані, за допомогою диспутів експертів із кількох країн, за відсутності викладача. Основну роль у здійсненні дистанційного навчання відіграють сучасні інформаційні технології [1].

Мета курсового проекту — є розробка елементів тренажеру з теми «Числення висловлювань» дистанційного навчального курсу «Математична логіка».

Об'єкт розробки — програма-тренажер для полегшення навчального процесу.

Предмет розробки — тренажер з теми «Числення висловлювань», який буде поєднано в дистанційний навчальний курс «Математична логіка».

Методи розробки — мова програмування Java, методи математичної логіки.

Дипломна робота складається з чотирьох розділів, а саме: постановка задачі, інформаційний огляд, теоретична частина. Структура роботи побудована так, що дає змогу логічного представлення матеріалу та розкриття теми роботи.

Для вирішення поставленої мети потрібно вирішити наступні задачі:

- проаналізувати існуючі тренажери, в яких розглянуто подібні до теми розробки;
- розробити алгоритм роботи тренажеру та його блок-схему;
- запрограмувати, протестувати та перевірити працездатність тренажера.

Для вирішення поставленої задачі було:

- проведено аналіз існуючого програмного забезпечення;
- розроблена алгоритмізація роботи тренажера;
- програмна реалізація тренажера.

Магістерська робота складається з вступу, чотирьох розділів (постановка задач, інформаційний огляд, теоретична частина, практична частина), списку використаних джерел і додатків.

Результатом виконання дипломної роботи є створення програмного забезпечення для тренажера з теми «Числення висловлення» за допомогою мови програмування *Java*. Практичне використання полягає в можливості використання розробленого програмного забезпечення при вивченні зазначеної теми.

1 ПОСТАНОВКА ЗАДАЧ

1.1 Основні вимоги до змісту роботи

При оформленні матеріалів дипломної роботи потрібно дотримуватись методичних рекомендацій щодо оформлення роботи студента зі спеціальності «Комп'ютерні науки». Зміст роботи повинен відповідати завданню затвердженому керівником та завідувачем кафедри[2].

Викладені матеріали повинні відповідати таким пунктам:

- чіткість, логічність та послідовність зробленого матеріалу;
- чітке формулювання думок, яке дозволить уникнути неясності матеріалу;
- рекомендації та пропозиції які потрібно проаналізувати .

Робота повинна містити:

- актуальність даної теми повинна відповідати вимогам сучасності;
- проаналізованість методів дослідження, та порівняльної характеристики;
- опис роботи розробленого програмного забезпечення;
- наявність блок-схеми програмного продукту;
- англomовний варіант тренажеру;

1.2 Постановка задачі для реалізації програми–тренажера з теми «Числення висловлювань»

Головною метою магістерського дипломного проекту є розробка програмного забезпечення для тренажера з теми «Числення висловлювань» дистанційного навчального курсу «Математична логіка». Розробка даного тренажеру дозволить добре засвоювати матеріал з даної теми.

Виходячи з викладеного вище, щоб досягнути поставленої мети потрібно виконати наступні завдання:

- проаналізувати схоже програмне забезпечення;
- розробити алгоритм тренажера та його блок-схему;
- програмно реалізувати тренажер, та перевірити його працездатність.

Тренажер повинен мати зрозумілий інтерфейс та бути легким у користуванні.

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Ефективність та недоліки дистанційного навчання

Популярність дистанційної освіти в останні роки різко зросла. Ця форма навчання є найбільш гнучкою та доступною для багатьох бажаючих отримати знання. Багато сказано на користь дистанційної освіти, і, мабуть, не менше про недоліки подібної форми навчання.

Серед суттєвих переваг дистанційної форми навчання можна відзначити наступні:

- можливість навчатися у будь-який час. Студент, який навчається дистанційно, може самостійно вирішувати, коли і скільки часу упродовж семестру йому приділяти на вивчення матеріалу. Він буде для себе індивідуальний графік навчання;
- можливість навчатися в будь-якому місці. Студенти можуть вчитися, не виходячи з дому чи офісу, перебуваючи у будь-якій точці планети. Щоб приступити до навчання, необхідний лише комп'ютер з доступом в Інтернет. Відсутність необхідності щодня відвідувати навчальний заклад - безсумнівний плюс для людей з обмеженими можливостями здоров'я, для проживаючих у важкодоступних місцевостях, батьків з маленькими дітьми;
- навчання без відриву від основної діяльності. Для навчання зовсім не обов'язково брати відпустку на основному місці роботи, виїжджати у відрядження. Також дистанційно можна навчатися на декількох курсах чи у декількох навчальних закладах одночасно;
- можливість навчатися у своєму темпі. Не обов'язково навчатися у тому ж темпі, що й інші студенти. Студент завжди може повернутися до вивчення більш складних питань, кілька разів подивитися відео-лекції, перечитати переписку з викладачем, а вже відомі йому теми може пропустити. Головне, успішно проходити проміжні та підсумкові атестації;

- доступність навчальних матеріалів. Доступ до всієї необхідної літератури відкривається студенту після реєстрації в системі дистанційного навчання, або він отримує навчальні матеріали електронною поштою. Зникає проблема нестачі чи відсутності підручників, навчальних посібників чи методичок;
- мобільність. Зв'язок з викладачами, репетиторами здійснюється різними способами: як on-line, так і off-line. Проконсультуватися з викладачем за допомогою електронної пошти іноді ефективніше та швидше, ніж призначити особисту зустріч при очному або заочному навчанні;
- навчання в спокійній обстановці. Проміжна атестація студентів дистанційних курсів проходить у формі on-line тестів. Тому в студентів менше причин для хвилювань. Виключається можливість суб'єктивної оцінки: на систему, яка перевіряє правильність відповідей на питання тесту, не вплине успішність студента з інших предметів, його соціальний статус та інші чинники;
- індивідуальний підхід. При традиційному навчанні викладачеві досить важко приділити необхідну кількість уваги всім студентам групи, підлаштуватися під темп роботи кожного. Використання дистанційних технологій підходить для організації індивідуального підходу. Крім того, що студент сам обирає собі темп навчання, він може оперативно отримати у викладачавідповіді на виникаючі питання;
- дистанційна освіта дешевша. Якщо порівнювати вартість навчання на заочній і дистанційній формах навчання, то дистанційна скоріш за все буде дешевшою. Студенту не доводиться оплачувати дорогу, проживання, а у випадку з зарубіжними вузами не потрібно витратитися на візу і закордонний паспорт;
- зручність для викладача. Вчителі, репетитори, викладачі, що займаються педагогічною діяльністю дистанційно, можуть приділяти увагу більшій кількості студентів і працювати, навіть перебуваючи у відрядженні чи на конференції за кордоном.

Разом з тим, дистанційне навчання не позбавлене і ряду недоліків:

- необхідна сильна мотивація. Практично весь навчальний матеріал студент-дистанційник освоює самостійно. Це вимагає достатньої сили волі, відповідальності і самоконтролю. Швидше за все, ніхто його підганяти чи заохочувати до навчання не стане. Підтримувати потрібний темп навчання без контролю з боку вдається не всім;
- нестача практичних вмінь та навиків. Досить проблемно якісно організувати дистанційне навчання за напрямками підготовки та спеціальностями, на яких передбачена велика кількість практичних занять. Навіть найсучасніші комп'ютерні тренажери не замінять майбутнім менеджерам «живої» практики;
- дистанційна освіта не підходить для розвитку комунікабельності. При дистанційному навчанні особистий контакт студентів один з одним і з викладачами мінімальний, а то і цілком відсутній. Тому така форма навчання не підходить для розвитку комунікабельності, впевненості, навичок роботи у команді;
- проблема ідентифікації студента. Поки найефективніший спосіб простежити за тим, чи студент самостійно здавав іспити чи заліки, - це відеоспостереження, що не завжди можливо. Тому на підсумкову атестацію студентам доводиться особисто приїжджати до вузу або його філії [3].

2.2 Огляд сучасних систем дистанційного навчання

Сьогодні широко використовується велика номенклатура систем дистанційного навчання та управління дистанційним навчанням як з відкритим кодом (умовно безкоштовних), так і платних, широкоживаних та вузькоорієнтованих.

ATutor – модульна система дистанційним керуванням навчанням з відкритим кодом. Поширюється на основі GNU General Public License. Для

установки необхідно мати комп'ютер з веб-сервером Apache 1.3.x, PHP версії > 4.2.0 та MySQL версій > 3.23.x і > 4.0.12 (версії 4.1.x і 5.x офіційно не підтримуються). Система розроблена із врахуванням доступності та можливістю адаптації за бажанням користувача. Щодо операційної системи сервера, обмежень немає – система є кросплатформеною.

Claroline – платформа дистанційного навчання та електронної діяльності з відкритим кодом. Аналогічно з ATutor, поширюється на основі GNU General Public License. Сумісна з такими операційними системами, як Linux, Mac і Windows. Забезпечує інтуїтивно простий інтерфейс для адміністрування. В основу організації Claroline LMS покладено концепцію просторів, пов'язаних з курсом чи педагогічною діяльністю. Кожен з просторів забезпечений інструментарієм для створення, організації та управління навчальними матеріалами; можливостями для забезпечення взаємодії між користувачами тощо.

Live@EDU – система дистанційного навчання, виконана з використанням технології Active Server Pages на платформі Microsoft. Для установки та коректної роботи системи серверна частина повинна бути забезпеченою ОС Microsoft Windows NT Server 4.0, базою даних Microsoft SQL Server 7.0 та Microsoft Internet Information Server 4.0. Клієнтська частина повинна мати установлену ОС, Lviv Polytechnic National University Institutional Repository яка забезпечує доступ до мережі Інтернет та браузер, що обслуговує протокол HTTP версії 3.0, а також програмне забезпечення для перегляду і створення лекційних матеріалів.

eFront є новим поколінням систем електронного навчання, яка об'єднує в собі функції системи управління навчанням та системи створення та управління навчальних матеріалів. Використовується для організації навчального процесу у навчальних закладах, а також для підвищення кваліфікації, атестації та відбору працівників у різномасштабних організаціях. Система побудована на трьох типах користувачів – Адміністратор, Викладач та Студент.

Moodle (Modular Object-Oriented Dynamic Learning Environment) – пакет модульного програмного забезпечення з відкритим кодом (ліцензія GNU GPL), який призначений для створення курсів дистанційного навчання та web-сайтів. Ця

програма управління дистанційним навчанням орієнтована на взаємодію між викладачем та студентом, також використовується для підтримки очних курсів. Moodle може бути встановленим на будь-який комп'ютер, який підтримує PHP та роботу із СУБД MySQL, PostgreSQL, Microsoft SQL Server; програмне забезпечення є кросплатформним.

SharePointLMS – система дистанційного навчання, розроблена на потужній багатофункціональній платформі MS Office SharePoint Server 2007. Є комплексним рішенням, яке об'єднує всіх користувачів (викладачі, студенти, адміністратори тощо) у єдиний інформаційно-навчальний простір та забезпечує інструментарій для спільної роботи. На відміну від Moodle, Claroline та

ATutor, система є платною. Використовується не лише навчальними закладами та центрами навчання, а й підприємствами, організаціями, державними структурами [4].

2.3 Огляд існуючих розробок

Під час аналізу існуючих розробок було виділено деякі програмні продукти:

На рис 2.1 зображено головну сторінку тренажера який розроблено на тему «Гرادієнтний метод».

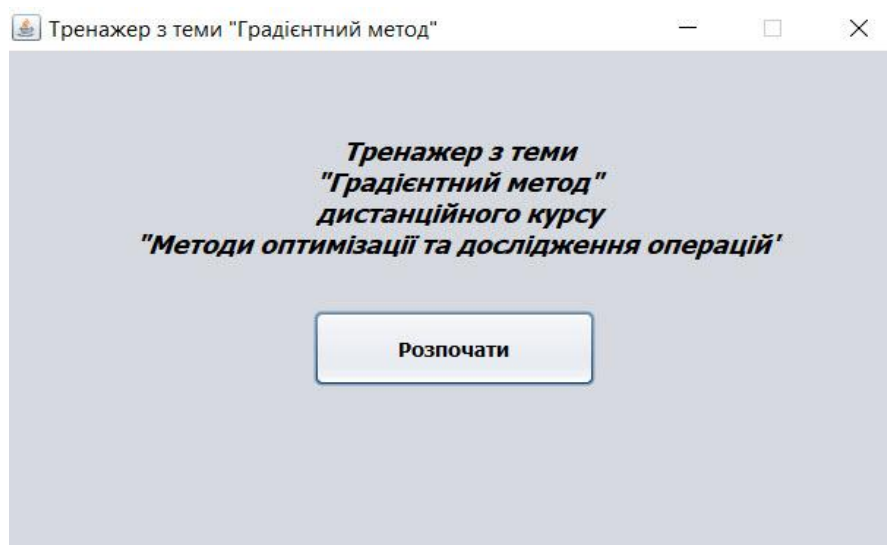


Рисунок 2.1 – головна сторінка з теми «Градiєнтний метод»

Під час запуску програми з'являється вікно завдання в якому зазначено обчислення методом найшвидшого спуску (рис 2.2).

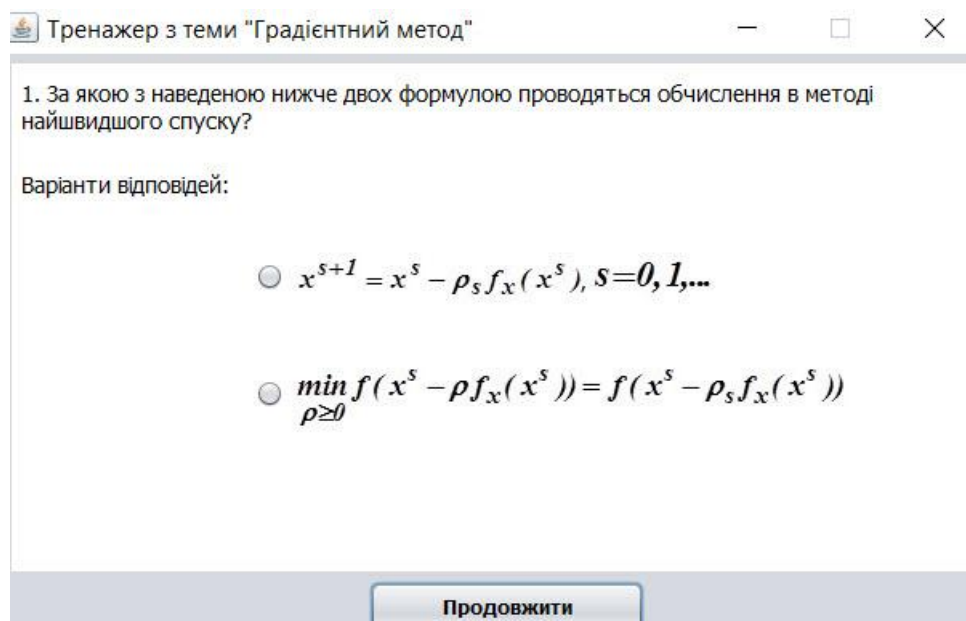


Рисунок 2.2 – вікно завдання з теми «Градiєнтний метод»

Також розглянемо тренажер з теми «Двоїстість в лінійному програмуванні», в якому відображено роботу програми, яка допомагає краще засвоїти матеріал с даної теми (рис 2.3).

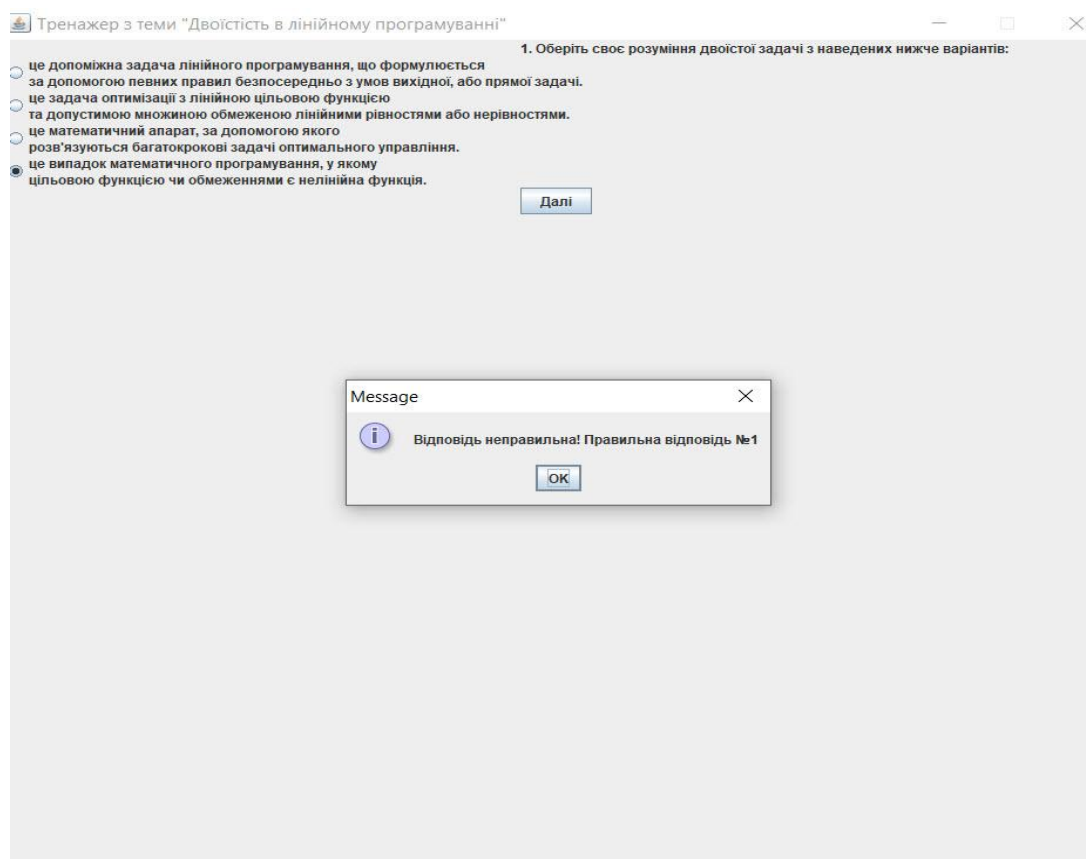


Рисунок 2.3 – вікно роботи програми з теми «Двоїстість в лінійному програмуванні»

2.4 Висновки за розділом 2

У результаті оглянутого матеріалу можна зробити такі висновки:

1. при порівнянні переваг та недоліків ДН було з'ясовано що таке навчання є досить перспективним;
2. при розробці тренажера існує досить багато програмного забезпечення.;
3. існуючи тренажери найчастіше використовують лише тільки тестовий формат.

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Алгоритмізація роботи тренажера

Перед тим, як навести алгоритм роботи тренажера, потрібно розглянути основні теоретичні відомості з теми «Числення висловлювань», а також

розглянути послідовність розв'язання прикладів, які покладені в основу його роботи.

3.2 Теоретичні відомості з теми «Числення висловлювань»

Числення висловлень – формальна система в математичній логіці, в якій формули, що відповідають висловленням, можуть утворюватись шляхом з'єднання простих висловлень із допомогою логічних операцій, та система правил виводу, які дозволяють визначати певні формули як «теореми» формальної системи [8].

Числення висловлювань має 11 схем аксіом:

- $A \Rightarrow (B \Rightarrow A)$;
- $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$;
- $(A \wedge B) \Rightarrow A$;
- $(A \wedge B) \Rightarrow B$;
- $A \Rightarrow (B \Rightarrow (A \wedge B))$;
- $A \Rightarrow (A \vee B)$;
- $B \Rightarrow (A \vee B)$;
- $(A \Rightarrow C) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \vee B \Rightarrow C))$;
- $A \Rightarrow (A \Rightarrow B)$;
- $(A \Rightarrow B) \Rightarrow ((A \Rightarrow \neg B) \Rightarrow \neg A)$;
- $A \vee \neg A$.

Числення висловлювань є несуперечливою, повною, розв'язною теорією. Усі ці три твердження доводяться в рамках самої логіки висловлювань.

У 1931 році Курт Гьодель надрукував свою знамениту працю: «Про формально нерозв'язні твердження у «Принципах математики» і спорідених системах». У цій праці Гьодель дослідив дві найбільш відомі формальні системи математики – систему «Принципи математики» Б. Рассела і А.Н. Уайтхеда, а також аксіоматичну теорію множин Цермело – Френкеля. У них було

формалізовано всі методи доведень, що використовуються в математиці, тобто доведення були зведені до сукупності деяких аксіом і правил виведення.

До публікації можна було припустити, що ці аксіоми і правила виведення достатні, щоб довести або спростувати будь-яке математичне твердження, яке можна формально виразити (записати) в цих системах. Виявилось, що таке припущення є хибним. У вказаних системах існують проблеми формальної арифметики, які є нерозв'язними, тобто які не можна ні довести, ні спростувати. І така ситуація є загальною, тобто не пов'язана зі специфікою формальних систем Рассела – Уайтхеда, Цермело – Френкеля чи інших систем.

Головний скарб праці Гьоделя – дві його знамениті теореми, названі теоремами про нерозв'язність (або про неповноту і несуперечливість). Ці теореми стосуються формальної системи, яка описує множину натуральних чисел і яку називають формальною арифметикою.

Перша теорема Гьоделя стверджує: якщо формальна арифметика несуперечлива, то вона неповна.

Друга теорема Гьоделя стверджує: несуперечливість формальної арифметики не можна довести засобами самої формальної арифметики.

Отримані результати було поширено на всі найбільш відомі формально-аксіоматичні системи: Рассела – Уайтхеда, Цермело – Френкеля, Гільберта і т.і. Стало зрозуміло, що кожна достатньо багата і несуперечлива система необхідно неповна. Така неповнота має принциповий характер, бо її неможливо усунути поступовим приєднанням до системи нових аксіом. А для доведення несуперечливості системи внутрішніх її засобів недостатньо. Тому потрібно використати сильніші, зовнішні відносно системи засоби.

Теореми Гьоделя засвідчують принципову обмеженість аксіоматичного методу. Кожна спроба викласти достатньо багату математичну теорію в рамки певної формальної системи призводить до тверджень, які неможливо ні довести, ні спростувати в межах цієї системи. Таким чином, Гьодель показав, що програма Гільберта повної формалізації математики є нездійсненою [11].

Доведенням в ЧВ (численні висловлень) називають послідовність формул, кожна з яких або є аксіомою, або одержується із попередніх за допомогою МР.

Пояснимо сказане прикладом доведення формули $(A \supset A)$.

В цьому записі із мовного оточення визначається, що A є формулою числення висловлень.

Доведенням є послідовність:

- $(A \supset (A \supset A))$ – аксіома 1;
- $((A \supset (A \supset A)) \supset ((A \supset ((A \supset A) \supset A)) \supset (A \supset A)))$ – аксіома 2;
- $((A \supset ((A \supset A)) \supset A) \supset (A \supset A))$ – modus ponens;
- $(A \supset ((A \supset A)) \supset A)$ – аксіома 1;
- $(A \supset A)$ — modus ponens.

Підставивши в схему аксіом 1 замість букви B та букви A формулу A , одержуємо вже аксіому $(A \supset (A \supset A))$, яка ставиться першою в доведенні потрібної формул [9].

Подібним чином, підставивши в схему аксіом 2 замість всіх букв формулу A одержуємо аксіому $((A \supset (A \supset A)) \supset ((A \supset ((A \supset A) \supset A)) \supset (A \supset A)))$.

Ця аксіома є імплікацією – із $(A \supset (A \supset A))$ випливає $((A \supset ((A \supset A) \supset A)) \supset (A \supset A))$. Припущення в цій імплікації вже є в нашому доведенні під номером 1. Тому можна застосувати правило відділення засновку (modus ponens) і одержати третій елемент доведення: $((A \supset ((A \supset A) \supset A)) \supset (A \supset A))$. Ця третя формула знову є імплікацією, засновком якої є формула $(A \supset ((A \supset A) \supset A))$. Ця формула є аксіомою – її можна одержати із першої схеми аксіом заміною B на $A \supset A$. Ця аксіома є четвертим елементом в доведенні. Тепер до третьої формули можна застосувати правило відділення засновку – засновок уже доведений. Таким чином, п'ятий елемент в доведенні є потрібна формула. Доведення закінчене.

Наведемо приклад доведення з використанням позалогічних аксіом і з використанням позначення \rightarrow для імплікації. Нехай нам відомі позалогічні аксіоми p , $\neg q$, і потрібно довести $\neg(p \rightarrow q)$:

- p – за умовою;

- $p \rightarrow ((p \rightarrow q) \rightarrow p)$ – аксіома 1;
- $(p \rightarrow q) \rightarrow p$ – МР до 2;
- $(p \rightarrow q) \rightarrow (p \rightarrow q)$ – переписана формула із попереднього прикладу. Отже

тут пропущена частина доведення. Доведені формули називають теоремами.

- $((p \rightarrow q) \rightarrow (p \rightarrow q)) \rightarrow (((p \rightarrow q) \rightarrow p) \rightarrow ((p \rightarrow q) \rightarrow q))$ — аксіома 2. В ній замість А підставлено $p \rightarrow q$, замість В підставлено p і замість С підставлено q [12].

Побудову формальної аксіоматичної теорії розпочинають з вичерпного опису мови формулювання висловлювань цієї теорії. Для стислості викладу подамо такий опис одночасно для логік нульового й першого порядку з указанням того, що відсутнє у мові логіки нульового порядку.

Мову логіки першого порядку будуються на основі:

- множини функціональних символів;
- множини предикатних символів.

Наприклад, до предикатних символів теорії множин належать такі:

- \in – читати: «належить до (множини) (як елемент)»;
- $=$ – читати: «дорівнює» або «збігається з».

З кожним функціональним і предикатним символом зв'язана арність (кількість аргументів). У логіці нульового порядку (численні висловлювань) функціональні та предикатні символи відсутні.

Додатково до цього використовують таке (далі сполучник «або» між символами означає, що у літературі зустрічаються усі перелічені позначення одного й того самого поняття):

- символи змінних, зазвичай $a, b, c, \dots, x, y, z, a_1, b_1$ і т.і. Передбачають зліченність символів, тобто можливість встановити взаємно однозначну відповідність між ними й підмножиною усіх натуральних чисел;
- оператор опису (дискрипції) ι — читати: «такий об'єкт, що ...» або «при якому».

Це поняття відсутнє у численні висловлювань;
логічні або пропозиційні зв'язки:

- \neg або \sim — читати: «не (справджується)»;
- \wedge або $\&$ — читати: «і»;
- \vee — читати: «або»;
- \Rightarrow або \rightarrow або \supset — читати: «впливає»;
- \Leftrightarrow або \leftrightarrow або \equiv або \sim — читати: «еквівалентно» або «(справджується)

тоді й лише тоді, коли (справджується)»;

- кома, дужки () для структурування запису;
- квантори, відсутні у численні висловлювань:
- \forall або (). — читати: «для всіх»;
- \exists або (E). — читати: «існує», у тому числі $\exists!$ — «існує єдиний».

3.3 Приклади та розв'язки завдань з теми «Числення висловлювань»

Завдання 1.

Чи є формулами числення висловлень такі вирази:

- 1) $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$;
- 2) $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$;
- 3) $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$;
- 4) $((((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))))$.

Розв'язання.

Формулами числення висловлень є вирази 1, 2, 4. Вираз 3 не є формулою, оскільки не має відповідності дужок.

Завдання 3.

Довести, що $a, b, a \rightarrow (b \rightarrow c) \vdash c$.

Розв'язання.

Формальна вивідність така:

F1: a ;

F2: b ;

F3: $a \rightarrow (b \rightarrow c)$;

F4: $MP(F1, F3) = b \rightarrow c$;

F5: $MP(F2, F4) = c$.

Завдання 3.

Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$.

Розв'язання.

Розбиття дужок на пари здійснюється індексами дужок: $(_1(2A \rightarrow B2) \rightarrow (_3(_4A \rightarrow C_4) \rightarrow (_5A \rightarrow (_6B \wedge C_6)_5)_3)_1)$. Область дії оператора вказано парою дужок: область дії першого оператора – це дужки 2-2, другого – дужки 1-1, третього – 4-4, четвертого – 3-3, п'ятого – 5-5, шестого – 6-6.

3.4 Алгоритм роботи тренажера

Після запуску тренажера, перед користувачем відкривається вікно, в якому можна обрати мову, на якій буде відображено інтерфейс тренажеру. Після вибору мови перед користувачем відкривається головне вікно тренажера, в якому можна почати виконувати тестові та практичні завдання з теми натиснувши кнопку «Розпочати». Розглянемо алгоритм роботи тренажера з тестовими і практичними завданнями більш детально.

Завдання 1.

Крок 1. Користувачу відображається перше запитання «Числення висловлювань – це?».

Нижче відображаються 3 варіанта відповідей:

- Формальна система в математичній логіці, в якій формули, що відповідають висловлюванням, можуть утворюватися шляхом з'єднання простих висловлень із допомогою логічних операцій, та система правил виводу, які дозволяють визначити певні формули як «теореми» формальної системи;
- Розділ математики, що вивчає мислення за допомогою числень, застосовуючи математичні методи та спеціальний апарат символів;
- Розділ математики, що вивчає систему логічних операцій над висловлюваннями.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 2.

Крок 2. Користувачу відображається друге запитання: «Числення висловлювань здійснює – ?».

Нижче відображаються 3 варіанта відповідей:

- Способи доведення на відношенні порядку, тобто на відношенні, яке є між причиною і наслідком;
- Формальне виведення істинності складних висловлювань та еквівалентні перетворення висловлювань;
- Будь-яку логічну операцію.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 3.

Крок 3. Користувачу відображається третє запитання : «Що означає множина \in ?

Нижче відображаються 3 варіанта відповідей:

- Належить до множини як елемент;
- Відношення еквівалентності;
- Виведення із множини.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 4.

Крок 4. Користувачу відображається четверте запитання : «Що стверджує перша теорема Гьюделя?»

Нижче відображаються 2 варіанта відповідей:

- Якщо формальна арифметика несуперечлива то вона неповна;
- Якщо формальна арифметика несуперечлива то вона повна.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 5.

Крок 5. Користувачу відображається п'яте запитання : «Що стверджує друга теорема Гьюделя?»

Нижче відображаються 3 варіанта відповідей:

- При побудові числення висловлювань можуть бути вибрані різні системи аксіом;
- Несуперечливість формальної арифметики не можна довести засобами самої формальної арифметики;
- Аксіом доведення нескінченно багато, їх стільки ж, скільки і формул.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 6.

Крок 6. Користувачу відображається шосте запитання : «Як поділяються аксіоми?»

Нижче відображаються 3 варіанта відповідей:

- Логічні і не логічні;
- Визначені і не визначені;
- Логічні і визначені.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 7.

Крок 7. Користувачу відображається сьоме запитання : «На основі чого будують мову логіки першого порядку?»

Нижче відображаються 3 варіанта відповідей:

- На теоремі дедукції;
- На множинах функціональних і предикатних символів;
- На відношенні порядку антисиметричності.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 8.

Крок 8. Користувачу відображається восьме запитання : «В численні висловлень визначають такі схеми аксіом:»

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 9.

Крок 9. Користувачу відображається дев'яте запитання : «Теорема дедукції дає:?»

Нижче відображаються 3 варіанта відповідей:

- Можливість виконати завдання за допомогою спеціальної формули предикатів;
- Новий перелік нових алгоритмів і аксіом;
- Нове правило виведення у логіці висловлювань, і тому є важливою при доведенні теорем і різних висловлювань, і формул.

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до кроку 10.

Крок 10. Користувачу відображається десяте запитання : «Як поділяються аксіоми?»

Нижче відображаються 3 варіанта відповідей:

- Рефлексивності, антисиметричності, транзитивності;

- Рефлексивності, транзитивності, паралельності;
- Антисиметричності, паралельності, рефлексивності.

Завдання 2.

Крок 1. Після тестової частини користувач переходить на 3 практичних завдання.

Користувачу відображається запитання :

Чи є формулами числення висловлень такі вирази:

- 1) $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$;
- 2) $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$;
- 3) $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$;
- 4) $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$.

Справа будуть зображені галочки де можна поставити правильні відповіді. Користувачеві потрібно буде вибрати 3 правильні відповіді. Якщо відповідь не правильна, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо користувач вибрав тільки одну чи дві відповіді з'явиться повідомлення «Виберіть три правильні відповіді». Після правильної відповіді користувач переходить до наступного завдання.

Крок 2. Користувачу потрібно довести формулу :

Довести, що $a, b, a \rightarrow (b \rightarrow c) \vdash c$.

В програмі буде зображено формула для якої потрібно буде підставити правильні значення. Для того щоб перейти до наступного запитання треба в кожную формулу підставити потрібні значення. Якщо користувач не підставить усі значення то висвітлиться вікно «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо ж користувач відповів як треба, то він переходить до наступного запитання.

Крок 3. Перед користувачем поставлено завдання:

Провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$.

Оскільки розбиття дужок на пари здійснюється індексами дужок, то користувачеві із запропонованої формули потрібно підставити правильно індекси.

Якщо користувач не підставить усі значення то висвітиться вікно «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо ж користувач відповів як треба, то він переходить до наступного запитання.

Приклад : $(_1(_2A \rightarrow B_2) \rightarrow (_3(_4A \rightarrow C_4) \rightarrow (_5A \rightarrow (_6B \wedge C_6)_5)_3)_1)$.

3.5 Розробка блок-схеми алгоритму роботи навчального тренажера

Для детальнішого огляду тренажеру з теми «Числення висловлень» потрібно продемонструвати його у вигляді блок-схеми.

Робота тренажеру починається з вибору української мови або англійської. Після вибору мови з'являється кнопка розпочати і після її натискання починається тестування.

Якщо користувач відповідає неправильно з'явиться повідомлення про те що відповідь не правильна. Якщо ж користувач відповів правильно він переходить до наступного питання

Блок-схема алгоритму роботи навчального тренажера з теми «Алгебра висловлень» зображена на рисунку 3.1.



Рис. 3.1 – Блок-схема роботи алгоритму тренажера

3.6 Обґрунтування вибору програмних засобів

Для розробки тренажера було обрано мову Java.

При виборі мови програмування необхідно врахувати такі вимоги:

- гнучкість. Java довела, що С – процедурний, керований вручну і залежить від платформи код – це не межа досконалості. Завдяки Java, все більше людей почали застосовувати об'єктно-орієнтоване програмування, яке зараз використовується повсюдно;
- аплети. Ще до появи JavaScript, в Java додали аплети – невеликі веб-програми, які надають інтерактивні елементи для візуалізації та навчання. Вони не використовуються ні для чого, крім простої анімації, однак аплети привернули увагу багатьох програмістів і підштовхнули їх до розробки HTML5, Flash і JavaScript;

– розробка через тестування. Java TDD – вже давно не експериментальна практика, а стандартний спосіб розробки програмного забезпечення. Введення JUnit 2000 року вважається одним з найбільших досягнень Java.

Java – це мова програмування загального призначення, який слід парадигмі об'єктно-орієнтованого програмування і підходу «Написати один раз і використовувати скрізь». Java використовується для настільних, мережових, мобільних і корпоративних додатків.

Програми на Java можуть бути трансльовані в байт-код, що виконується на віртуальній java-машині (JVM) – програмою, обробній байт-код і передавальній інструкції обладнанню, як інтерпретатор, але з тією відмінністю, що байт-код, на відміну від тексту, обробляється значно швидше.

Мова Java зародився як частина проекту створення передового програмного забезпечення для різних побутових приладів. Реалізація проекту була почата на мові C ++, але незабаром виник ряд проблем, найкращим засобом боротьби з якими була зміна самого інструмента - мови програмування. Стало очевидним, що необхідний платформо-незалежний мова програмування, що дозволяє створювати програми, які не доводилося б компілювати окремо для кожної архітектури і можна було б використовувати на різних процесорах під різними операційними системами.

Мова Java потрібен для створення інтерактивних продуктів для мережі Internet. Фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, схожий з C і C ++. В Java використовуються практично ідентичні угоди для оголошення змінних, передачі параметрів, операторів і для управління потоком виконання коду. В Java додані всі хороші риси C ++ [5].

3.7 Висновки за розділом 3

У результаті виконання теоретичної частини роботи можна зробити наступні висновки:

1. Для реалізації тренажеру з теми «Числення висловлювань» було проаналізовано багато лекційного матеріалу;
2. Розписано алгоритм програми а також його блок-схема.
3. Обґрунтовано основні вимоги для вибору ПЗ.

4 ПРАКТИЧНА ЧАСТИНА

4.1 Опис процесу програмної реалізації

Основний код тренажера з теми «Числення висловлень» написаний в java за допомогою мови розмітки XML. Програма має спроектовані вікна які зв'язані між собою та мають переходи серед завдань.

Основний код програми написаний в файлах java який в той час посилається на інтерфейс з розширенням fxml.

Приклад підключення бібліотек:

```

import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;

@FXML
private Label Label1

@FXML
private Button Button 1

@FXML

private Button Button 1

@FXML

void initialize(){
    assert Label 1 != null : "fx:id=\"NextTask\" was not injected: check your FXML file
'Program_Interface.fxml'.";
    assert Button 1 != null : "fx:id=\"NextTask\" was not injected: check your FXML file
'Program_Interface.fxml'.";
    assert Button 1 != null : "fx:id=\"NextTask\" was not injected: check your FXML file
'Program_Interface.fxml'.";

```

Тести створені за допомогою спеціальних кнопок CheckBox, RadioButton та ComboBox. При натисканні користувач повинен обрати правильну відповідь і програма перевіряє правильно це чи ні. Якщо ж відповідь не правильна з'являється спеціальне вікно про те що відповідь дана не правильно.

4.2 Опис програми

Для налаштування основних форм програми потрібно імпортувати спеціальні бібліотеки `control` та `layout`.

Приклад коду:

```
package sample;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.geometry.Orientation;
import javafx.geometry.Pos;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.layout.FlowPane;
import javafx.scene.layout.StackPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.fxml.FXMLLoader;
```

Далі потрібно розмістити робоче вікно *Pane*. При кодуванні розмічаємо висоту, ширину, та java файл який буде взаємодіяти с інтерфейсом.

Приклад коду:

```
<Pane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="303.0" prefWidth="600.0" style="-fx-background-color: #000000;"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="sample.Controller22ENG">
  <children>
    <ImageView fitHeight="393.0" fitWidth="600.0" layoutY="-3.0" pickOnBounds="true"
preserveRatio="true">
      <image>
        <Image url="@quest.jpg" />
      </image>
    </ImageView>
    <HBox layoutX="421.0" layoutY="152.0" prefHeight="25.0" prefWidth="53.0" style="-fx-background-
```

```

color: #ffffff;">
    <children>
        <RadioButton fx:id="RadioButton3" mnemonicParsing="false" prefHeight="25.0" prefWidth="49.0"
text="3-3" wrapText="true" />
    </children></HBox>
    <HBox layoutX="267.0" layoutY="14.0" prefHeight="25.0" prefWidth="308.0" style="-fx-background-
color: #ffffff;">
        <children>
            <Label prefHeight="20.0" prefWidth="78.0" text="Prove that">
                <font>
                    <Font name="System Bold Italic" size="14.0" />
                </font>
            </Label>
            <ImageView fitHeight="29.0" fitWidth="207.0" pickOnBounds="true" preserveRatio="true">
                <image>
                    <Image url="@quest13.png" />
                </image>
            </ImageView>
        </children></HBox>
        <HBox layoutX="337.0" layoutY="152.0" prefHeight="25.0" prefWidth="46.0" style="-fx-background-
color: #ffffff;">
            <children>
                <RadioButton fx:id="RadioButton2" mnemonicParsing="false" prefHeight="25.0" prefWidth="53.0"
text="2-2" wrapText="true">
                    <font>
                        <Font size="13.0" />
                    </font>
                </RadioButton>
            </children></HBox>
            <HBox layoutX="254.0" layoutY="152.0" prefHeight="25.0" prefWidth="53.0" style="-fx-background-
color: #ffffff;">
                <children>
                    <RadioButton fx:id="RadioButton1" mnemonicParsing="false" prefHeight="25.0" prefWidth="56.0"
text="1-1" wrapText="true">
                        <font>
                            <Font size="14.0" />
                        </font></RadioButton>
                </children></HBox>
                <Button fx:id="Button" layoutX="535.0" layoutY="264.0" mnemonicParsing="false" text="Next" />
                <HBox layoutX="267.0" layoutY="39.0" prefHeight="18.0" prefWidth="308.0" style="-fx-background-
color: #ffffff;">
                    <children>
                        <Label prefHeight="18.0" prefWidth="258.0" text="Let's break souls into indexes:">
                            <font>
                                <Font name="System Bold Italic" size="11.0" />
                            </font>
                        </Label>
                    </children>
                </HBox>
                <HBox layoutX="260.0" layoutY="201.0" prefHeight="20.0" prefWidth="53.0" style="-fx-background-
color: #ffffff;">
                    <children>
                        <RadioButton fx:id="RadioButton4" mnemonicParsing="false" prefHeight="20.0" prefWidth="49.0"
text="4-4" wrapText="true" />
                    </children>
                </HBox>
                <HBox layoutX="334.0" layoutY="200.0" prefHeight="25.0" prefWidth="53.0" style="-fx-background-

```

```

color: #ffffff; ">
    <children>
        <RadioButton fx:id="RadioButton5" mnemonicParsing="false" prefHeight="25.0" prefWidth="54.0"
text="5-5" wrapText="true">
            <font>
                <Font size="13.0" />
            </font>
        </RadioButton>
    </children>
</HBox>
<HBox layoutX="267.0" layoutY="57.0" prefHeight="25.0" prefWidth="308.0" style="-fx-background-
color: #ffffff; ">
    <children>
        <ImageView fitHeight="30.0" fitWidth="300.0" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@quest13_().png" />
            </image>
        </ImageView>
    </children>
</HBox>
<HBox layoutX="267.0" layoutY="82.0" prefHeight="18.0" prefWidth="308.0" style="-fx-background-
color: #ffffff; ">
    <children>
        <Label prefHeight="18.0" prefWidth="258.0" text="Select a couple of brackets for the sixth operator:">
            <font>
                <Font name="System Bold Italic" size="11.0" />
            </font>
        </Label>
    </children>
</HBox>
<HBox layoutX="421.0" layoutY="199.0" prefHeight="25.0" prefWidth="53.0" style="-fx-background-
color: #ffffff; ">
    <children>
        <RadioButton fx:id="RadioButton6" mnemonicParsing="false" prefHeight="25.0" prefWidth="54.0"
text="6-6" wrapText="true">
            <font>
                <Font size="13.0" />
            </font>
        </RadioButton>
    </children>
</HBox>
</children>
</Pane>

```

Оскільки *Pane* є головним елементом, то увесь інтерфейс буде розмічени в класі *children*.

К кодi також знаходяться *imageView*, *Button*, *Label*, та *RadioButton*.

На прикладі елемента *label* розглянемо його властивості :

- положення;
- розмір.

Приклад коду:

```

<Label prefHeight="18.0" prefWidth="258.0" text="Виберіть пару дужок шостого оператора:">
    <font>

```



```

<Font name="System Bold Italic" size="11.0" />
</font>
</Label>

```

Далі програмуємо основний файл java який відповідає за роботу усієї програми. Спочатку підключаємо бібліотеки і імпортуємо класові файли.

Приклад коду:

```

package sample;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.geometry.Orientation;
import javafx.geometry.Pos;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.CheckBox;
import javafx.scene.control.Label;
import javafx.scene.layout.FlowPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

```

Далі описуємо клас за допомогою команди *public class*. Підключаємо все елементи fxml, та з нового рядка оголошуємо сам елемент privat.

Приклад коду:

```

public class Controller11 {
    @FXML
    private CheckBox CheckBox2;
    @FXML
    private CheckBox CheckBox1;
    @FXML
    private Button Button;
    @FXML
    private CheckBox CheckBox4;
    @FXML
    private CheckBox CheckBox3;
}

```

Далі програмуємо натискання кнопки яка відповідає за правильні відповіді за допомогою оператора *if*.

Приклад коду:

```

if (RadioButton2.isSelected())
    else
}.

```

Якщо користувач відповів правильно він переходить до наступного завдання до наступного завдання. Якщо користувач відповів неправильно з'явиться спеціальне вікно про те що відповідь не правильна.

Приклад коду:

```
if (RadioButton2.isSelected()){
    {
        RadioButton2.getScene().getWindow().hide();
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("18.fxml"));
        try {
            loader.load();
        } catch (IOException e) {
            e.printStackTrace();
        }
        Parent root = loader.getRoot();
        Stage Practic = new Stage();
        Practic.setTitle("Числення висловлень");
        Practic.setScene(new Scene(root));
        Practic.show();
    }
}
else if (RadioButton4.isSelected()){
    Label lb = new Label("Не вірно!\n ");
    Button bt = new Button ("Спобувати ще!");
    bt.setOnAction(new EventHandler<ActionEvent>() {
        @Override
        public void handle(ActionEvent event) {
            bt.getScene().getWindow().hide();
        }
    });
    FlowPane root = new FlowPane(Orientation.VERTICAL, lb, bt);
    root.setAlignment(Pos.CENTER);
    Scene secondScene = new Scene(root, 500, 500 );
    // New window (Stage)
    Stage newWindow = new Stage();
    newWindow.setTitle("Second Stage");
    newWindow.setScene(secondScene);
    // Specifies the modality for new window.
    newWindow.initModality(Modality.APPLICATION_MODAL);
    newWindow.show();
}
```

4.3 Перевірка валідності

Для перевірки працездатності програми тестувалися всі вікна програми. При запуску програмного продукту з'явиться стартове вікно, де запропонують обрати мову тренажера (рис. 4.3). Тестування було проведено на українській мові, на англійській мові можливості тренажеру нічим не відрізняються (рис 4.4).



Рисунок 4.3 – Вікно вибору мови

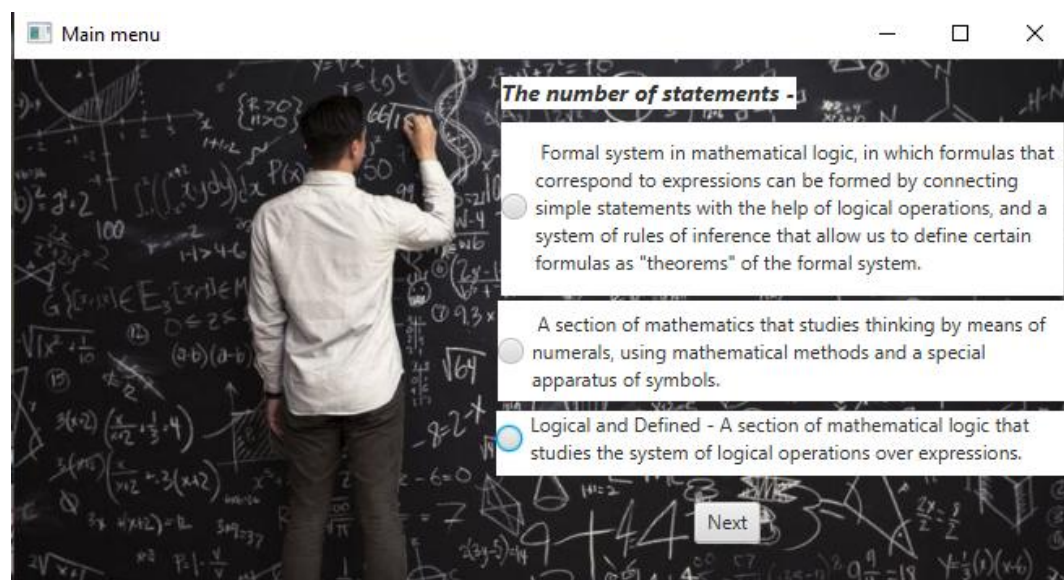


Рисунок 4.4 – Вибір англійської версії

На рис. 4.5 зображено головне меню з кнопкою для роботи тестів.



Рис. 4.5 – Головне вікно

При натисканні кнопки «Розпочати» відображується перше запитання з теми «Числення висловлювань». Перше запитання «Числення висловлювань – це?» (рис. 4.6)

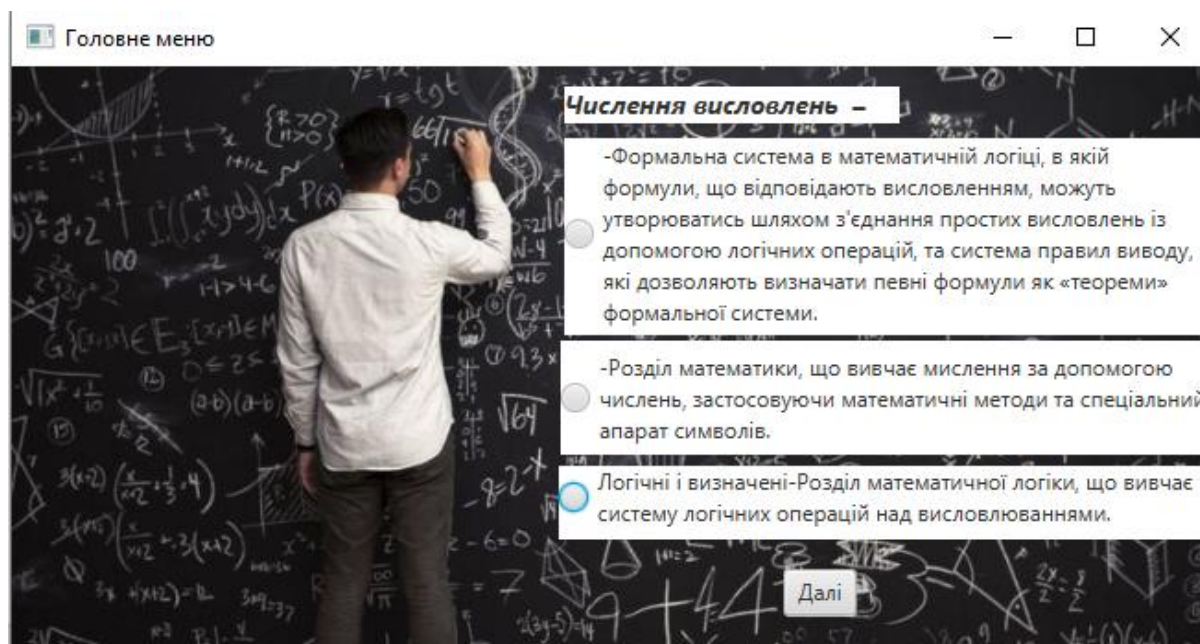


Рис. 4.6 – Перше запитання

Якщо користувач відповідає правильно він переходить до наступного запитання «Числення висловлювань здійснює – ?» (рис. 4.8). Якщо відповідь не

правильна з'явиться вікно про неправильність відповіді і запропонує користувачеві спробувати ще раз (рис. 4.7).

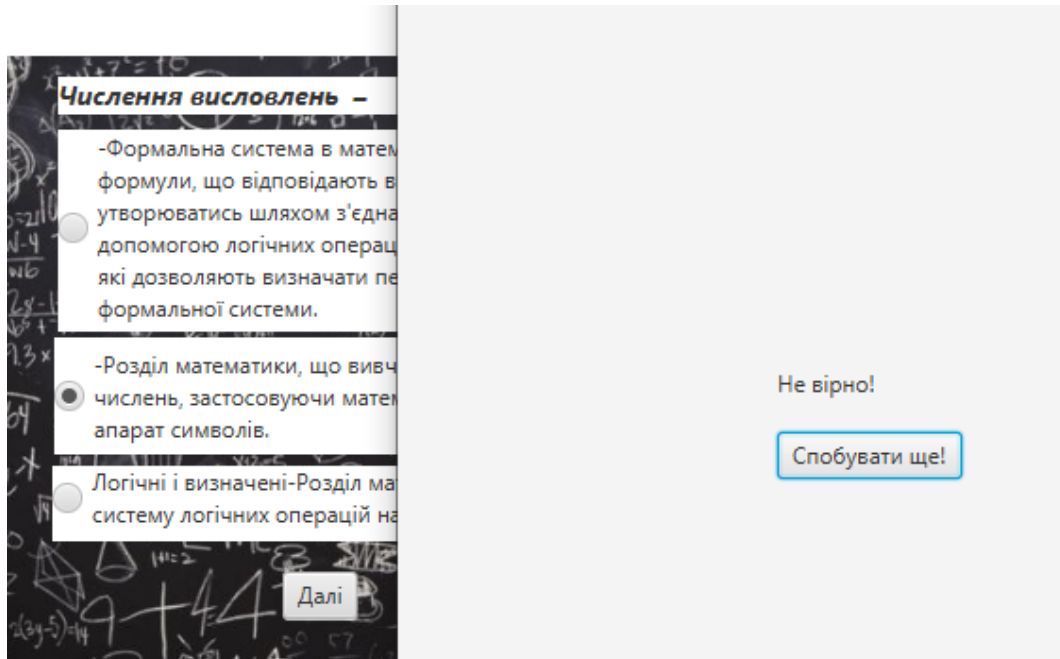


Рис. 4.8 – Повідомлення про помилку

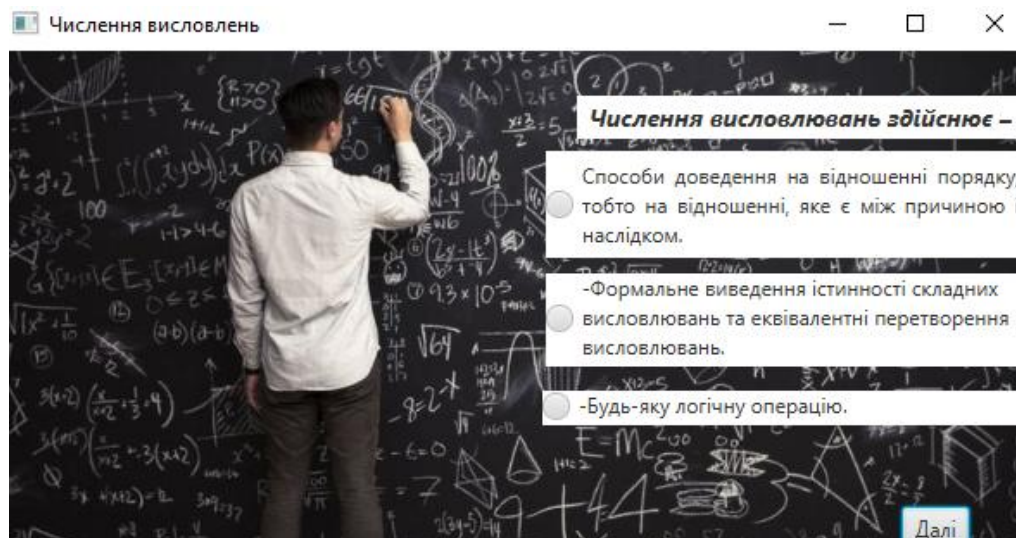


Рис. 4.7 – Друге запитання

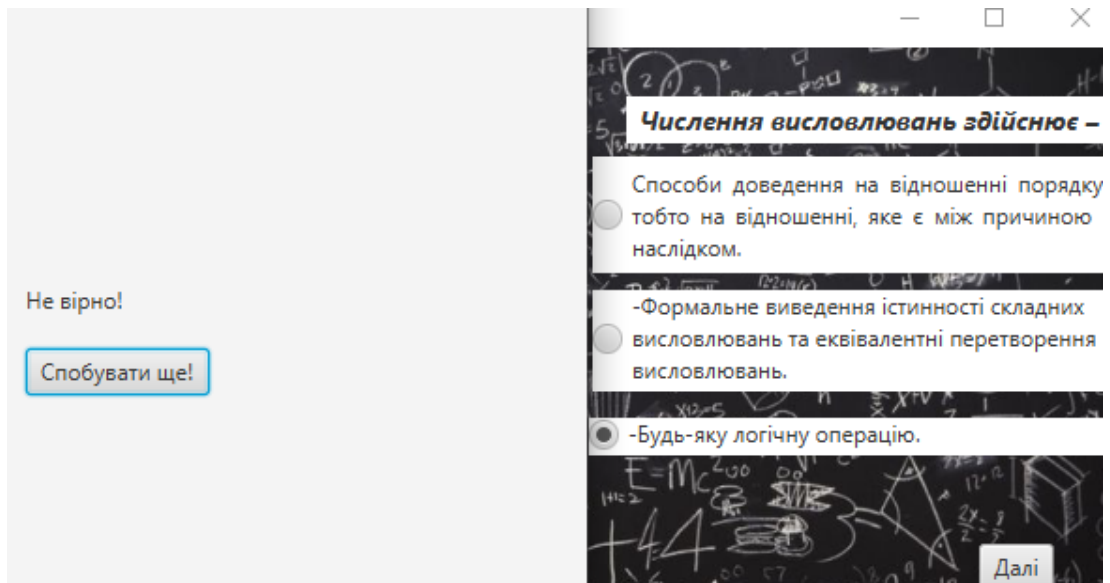


Рис. 4.9 – Повідомлення про помилку

Далі користувачу відображається третє запитання : «Що означає множина \in ?» (рис. 4.10). Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.11), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить наступного запитання.



Рис. 4.10 – Третє запитання

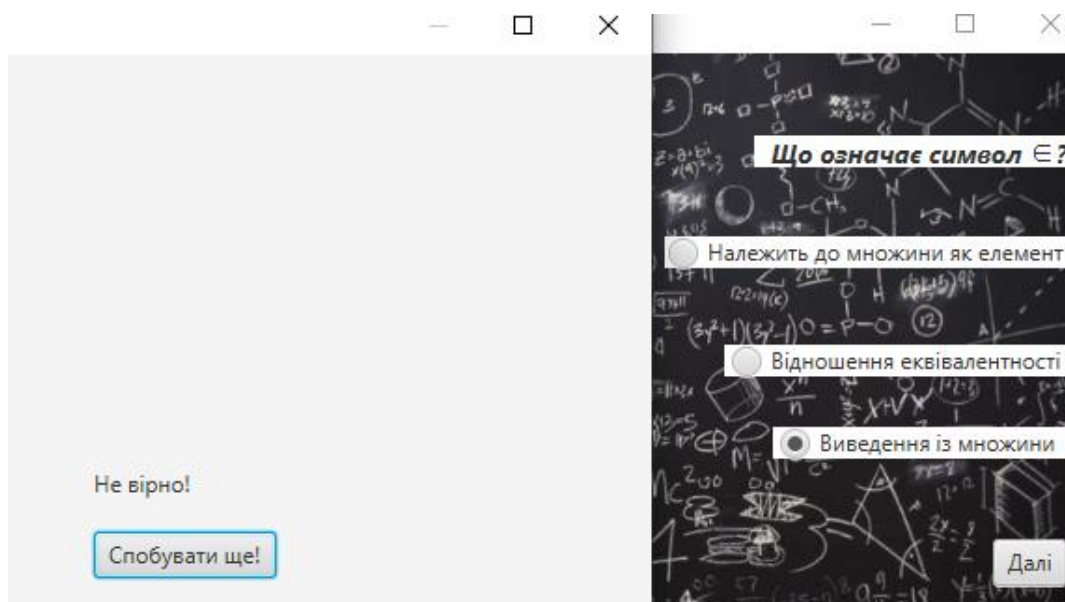


Рис. 4.11 – Повідомлення про помилку

Далі користувачу відображається четверте запитання : «Що стверджує перша теорема Гьюделя?» (рис.4.12).

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис. 4.13), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до наступного запитання.

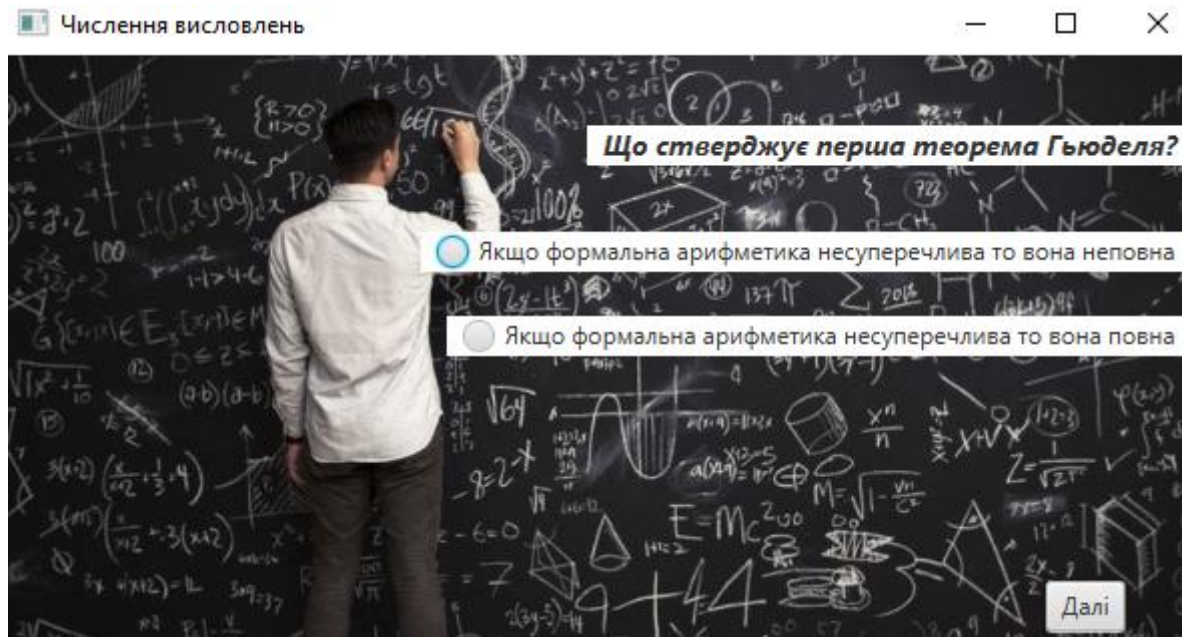


Рис. 4.12 – Четверте запитання



Рис. 4.13 – Повідомлення про помилку

Далі користувачу відображається п'яте запитання : «Що стверджує друга теорема Гьюделя?» (рис.4.14). Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.15), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до наступного запитання.

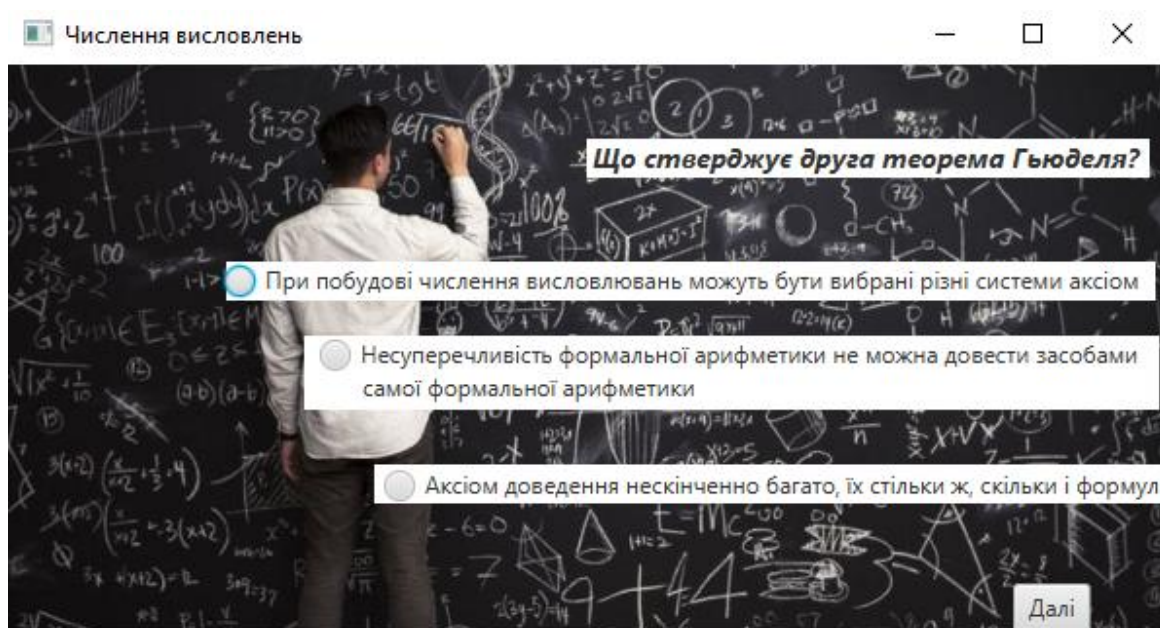


Рис. 4.14 – П'яте запитання

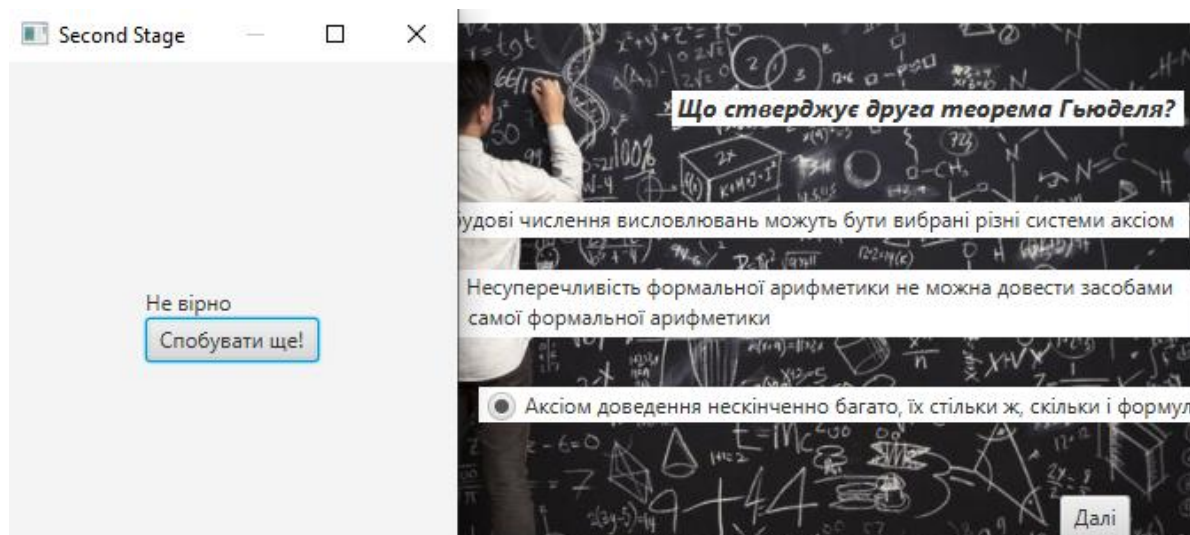


Рис. 4.15 – Повідомлення про помилку

Далі користувачеві відображається шосте запитання : «Як поділяються аксіоми?» (рис.4.16).

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.17), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач наступного запитання.



Рис. 4.16 – Шосте запитання

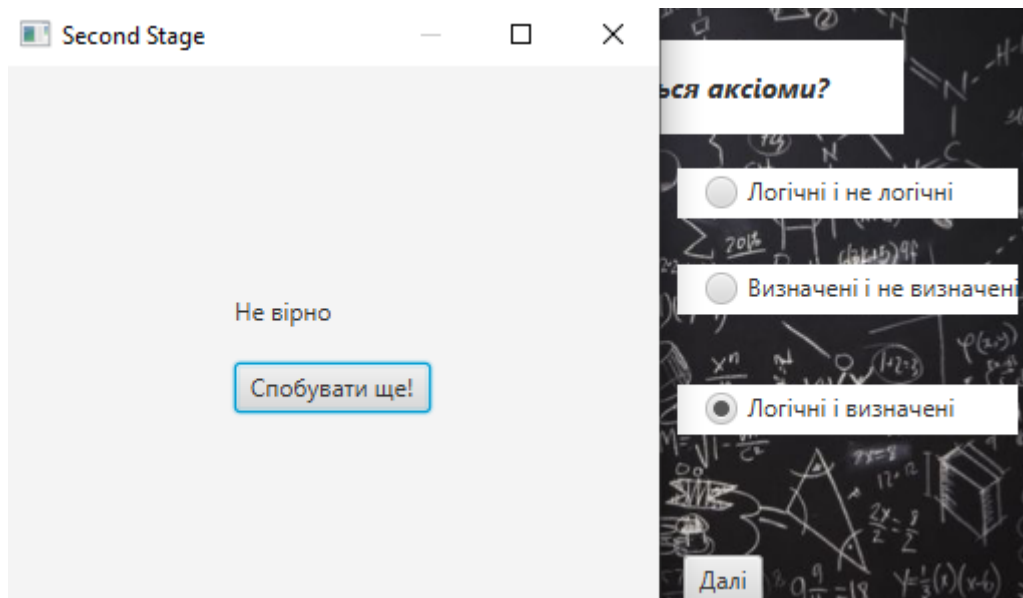


Рис. 4.17 – Повідомлення про помилку

Далі користувачу відображається сьоме запитання : «На основі чого будують мову логіки першого порядку?» (рис.4.18).

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.19), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до наступного запитання.

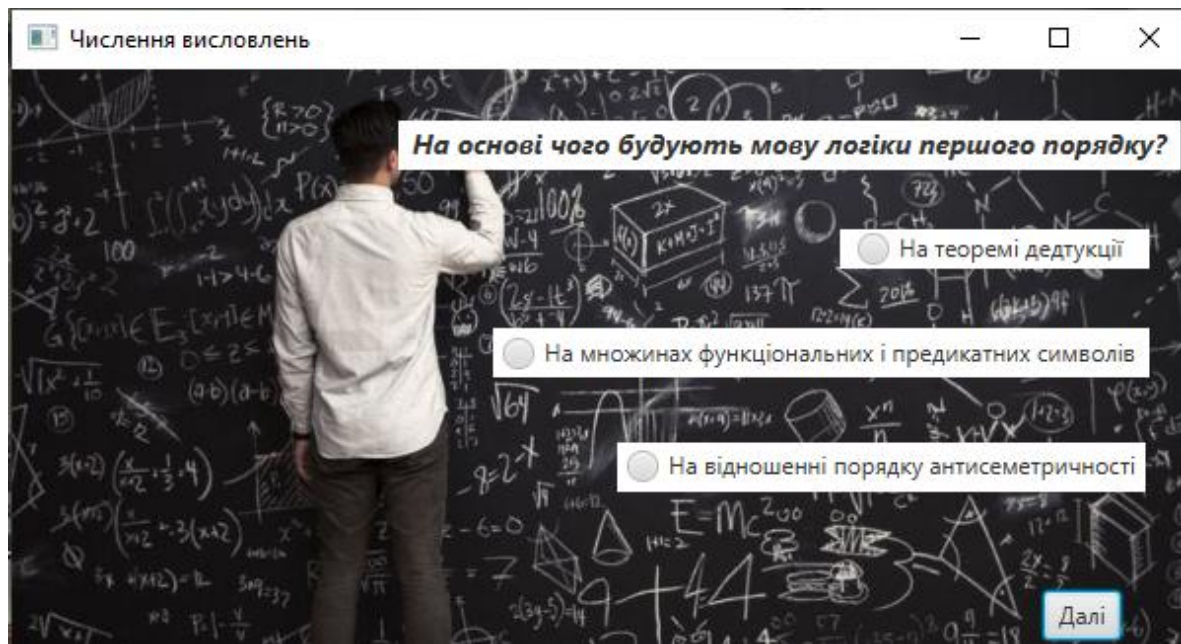


Рис. 4.18 – Сьоме запитання

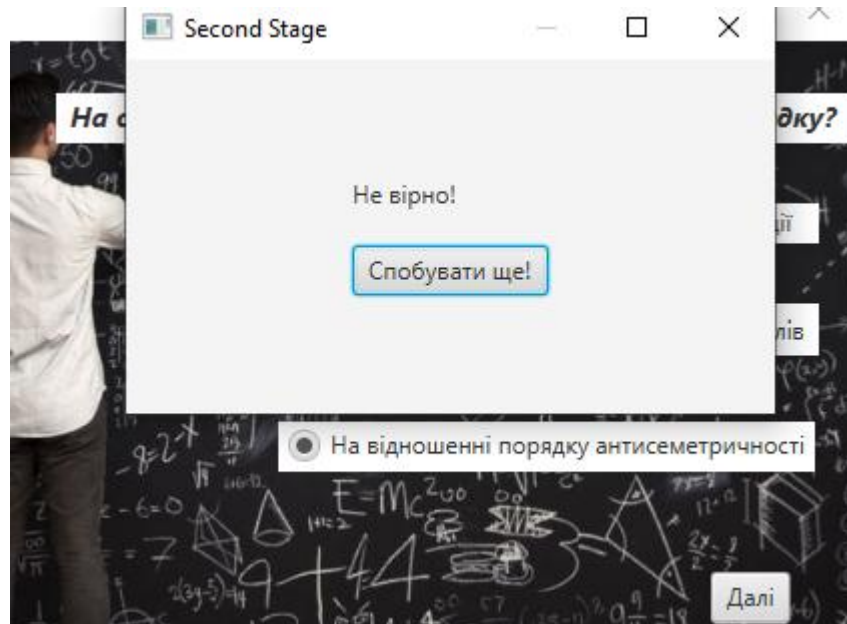


Рис. 4.19 – Повідомлення про помилку

Далі користувачеві відображається восьме запитання : «В численні висловлень визначають такі схеми аксіом:» (рис 4.20).

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.21), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до наступного запитання.

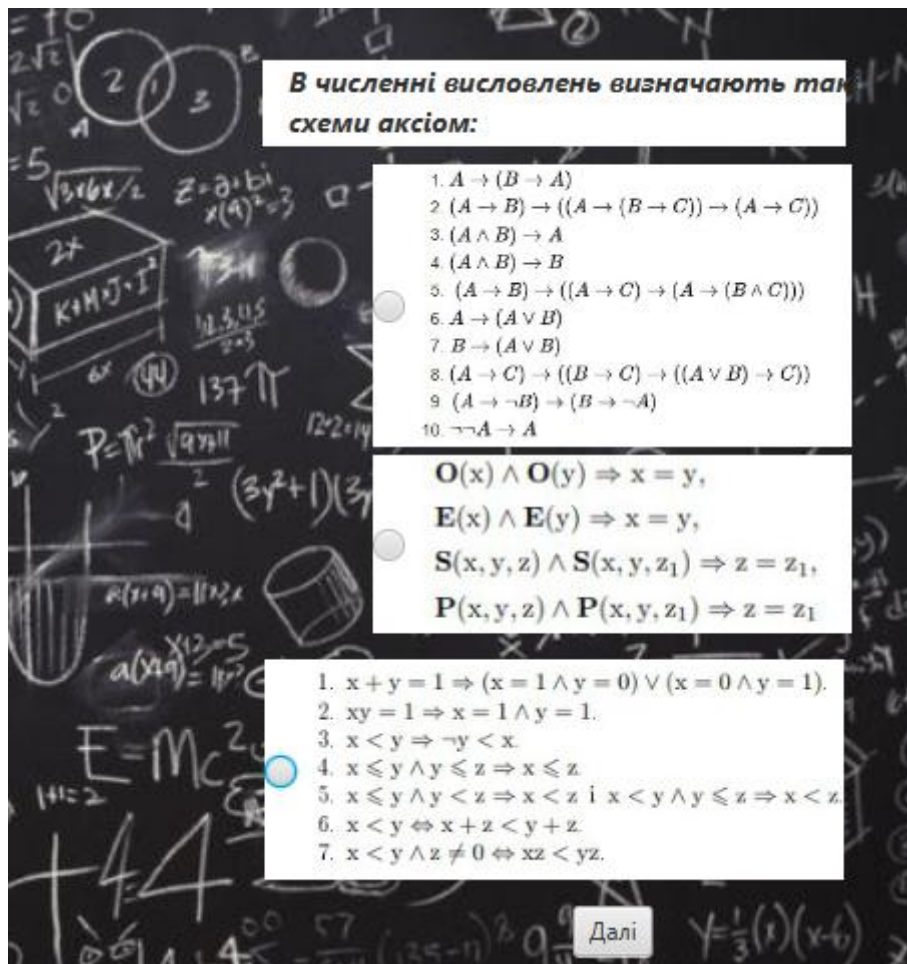


Рис. 4.20 – Восьме запитання

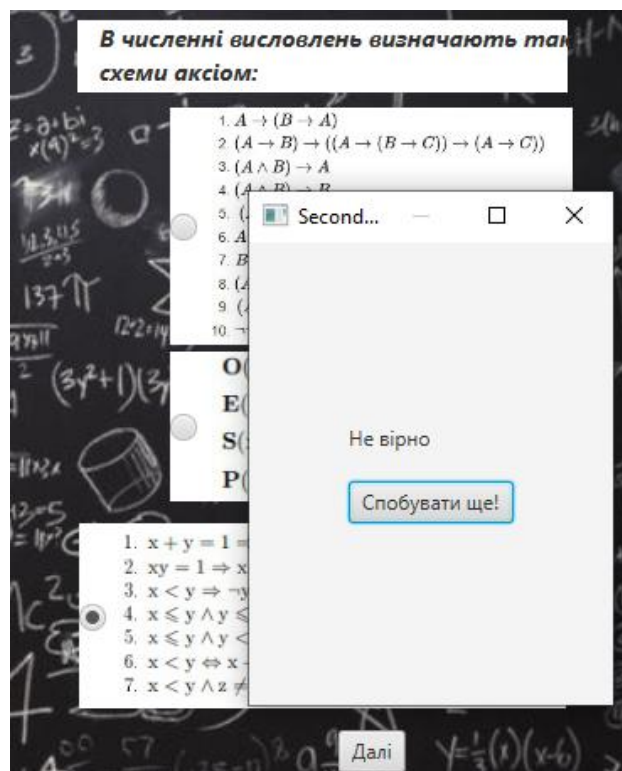


Рис. 4.21 – Повідомлення про помилку

Далі користувачу відображається дев'яте запитання : «Теорема дедукції дає:?» (рис. 4.22).

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.23), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до наступного запитання.

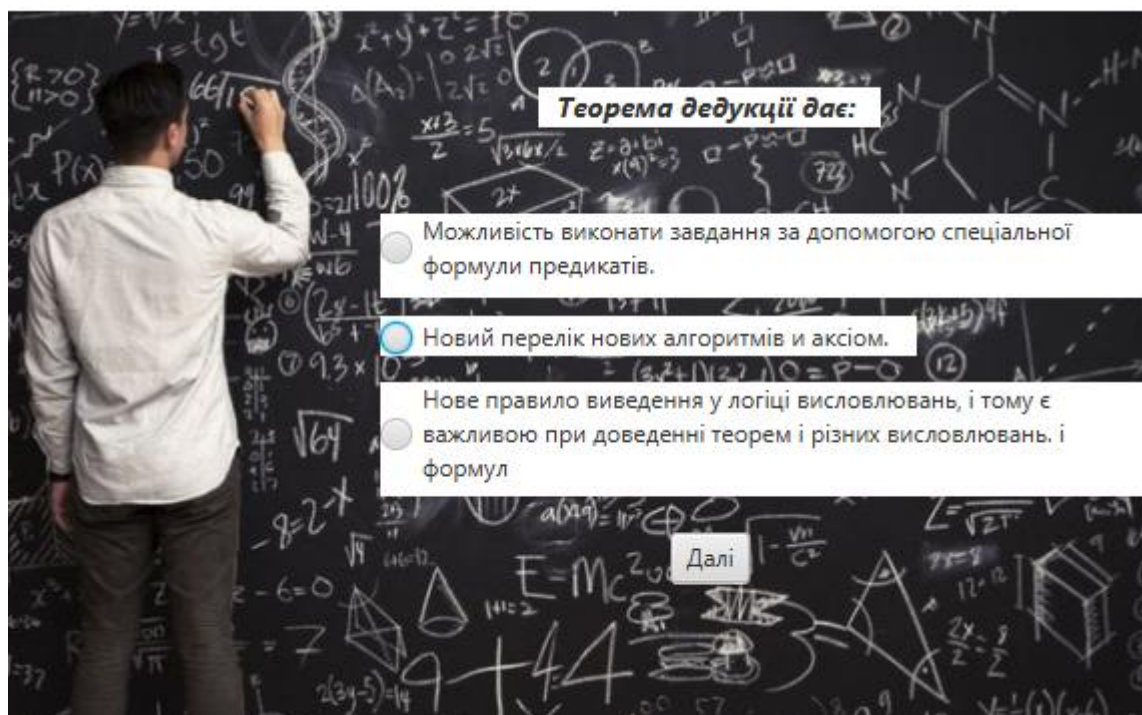


Рис. 4.22 – Дев'яте запитання

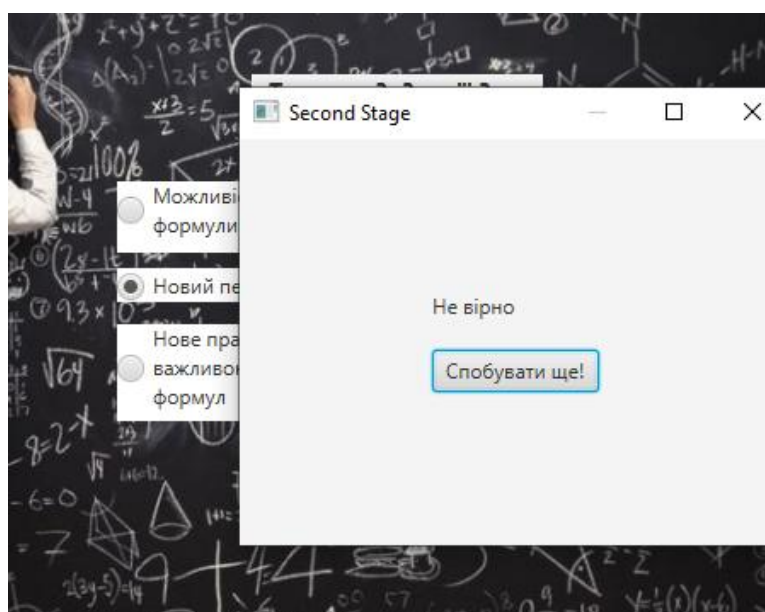


Рис. 4.23 – Повідомлення про помилку

Далі користувачу відображається десяте запитання : «Як поділяються аксіоми?» (4.24).

Користувачу потрібно серед запропонованих варіантів відповідей обрати 1 правильну відповідь. Якщо користувач вибере не правильну відповідь, з'явиться повідомлення «Не вірно» (рис.4.25), та запропонує користувачеві спробувати ще раз. Якщо вибрана відповідь правильна, то користувач переходить до практичних запитань.

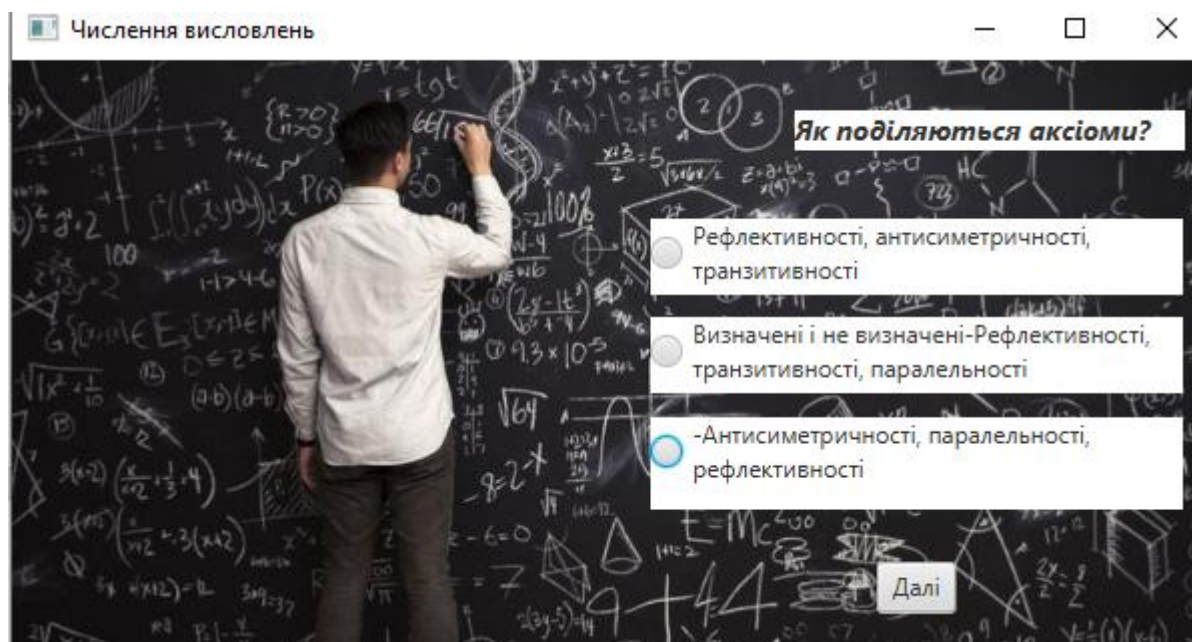


Рис. 4.24 – Десяте запитання

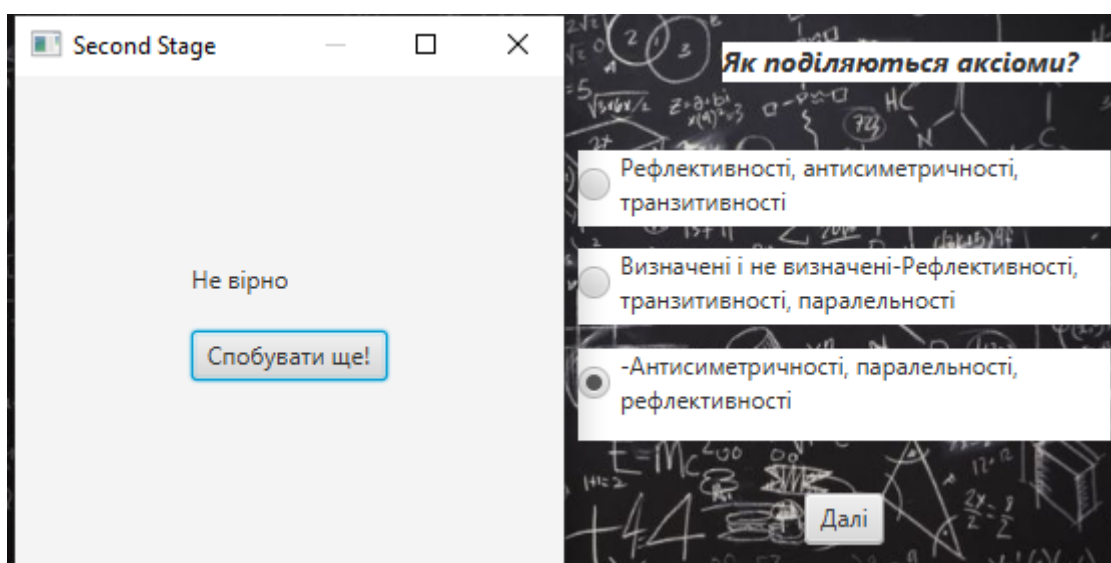


Рис. 4.25 – Повідомлення про помилку

Після тестової частини користувач переходить на 3 практичних завдання.

Користувачу відображається запитання :

Чи є формулами числення висловлень такі вирази (рис.2.26):

- 1) $((A \vee ((\neg B) \rightarrow C)) \rightarrow (\neg A))$;
- 2) $((A \wedge B) \vee (C \wedge D) \rightarrow (A \wedge C))$;
- 3) $(A \rightarrow C) \leftrightarrow ((\neg C) \rightarrow (\neg A))$;
- 4) $((A \wedge B) \wedge C) \rightarrow ((B \vee C) \vee (\neg D))$.

Користувачеві потрібно буде вибрати 3 правильні відповіді. Якщо відповідь не правильна, з'явиться повідомлення «Не вірно», та запропонує користувачеві спробувати ще раз. Якщо користувач вибрав тільки одну чи дві відповіді з'явиться повідомлення «Виберіть три правильні відповіді» (рис.2.28). Після правильної відповіді користувач переходить до наступного завдання.

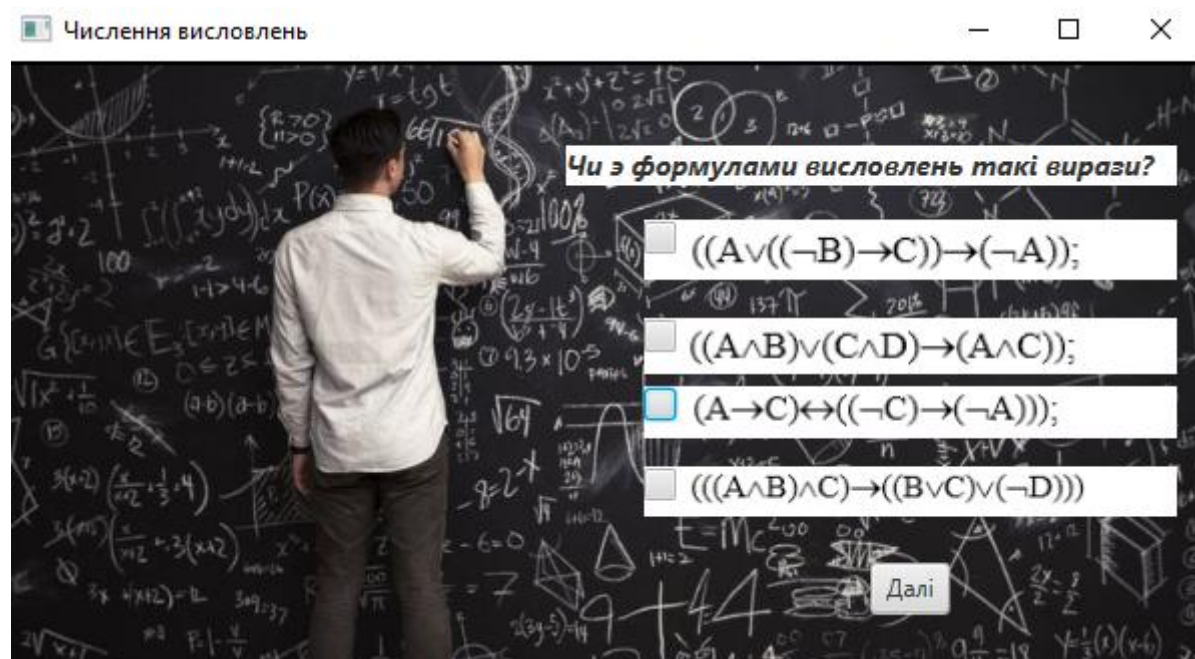


Рис. 4.26 – Перше практичне завдання

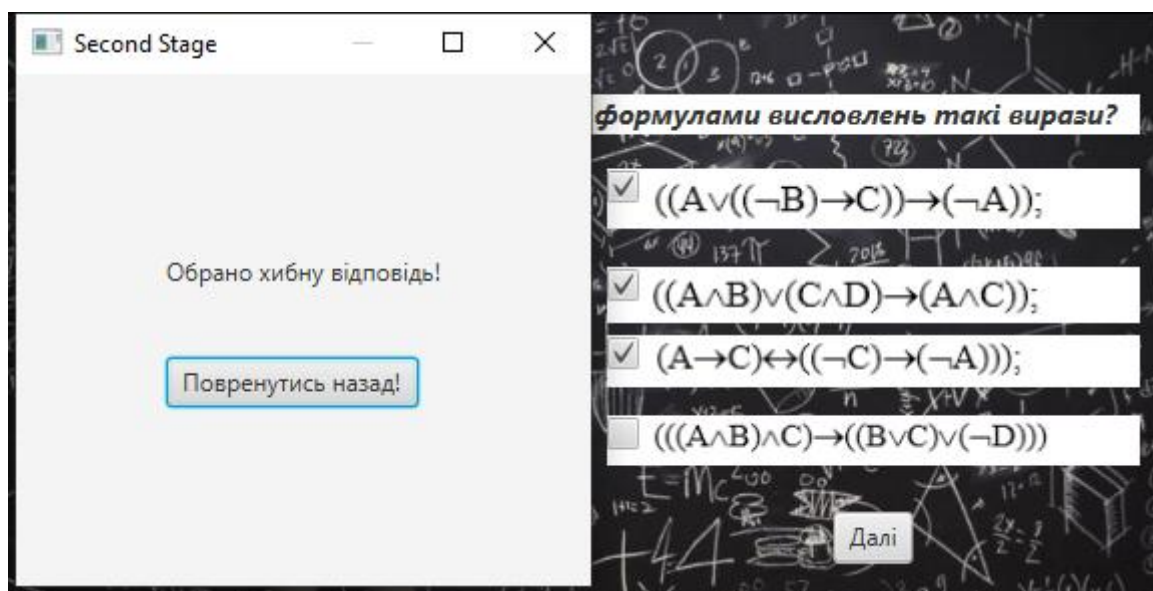


Рис. 4.27 – Повідомлення про помилку

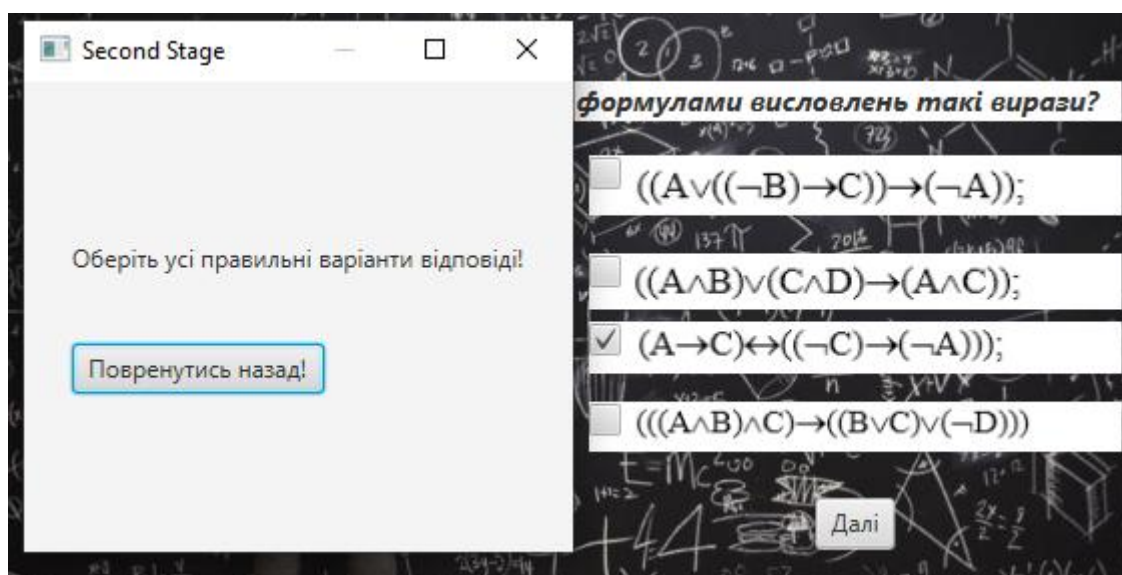


Рис. 4.28 – Повідомлення про те що вибрані не все відповіді

Далі користувачу потрібно довести формулу :

Довести, що $a, b, a \rightarrow (b \rightarrow c) \vdash c$ (рис.2.29).

В програмі буде зображено формула для якої потрібно буде підставити п'ять правильних значень. Для того щоб перейти до наступного запитання треба в кожну формулу підставити потрібні значення. Якщо користувач не підставить усі значення то висвітиться вікно «Не вірно» (рис.2.30), та запропонує користувачеві спробувати ще раз. Якщо ж користувач відповів як треба, то він переходить до наступного запитання.

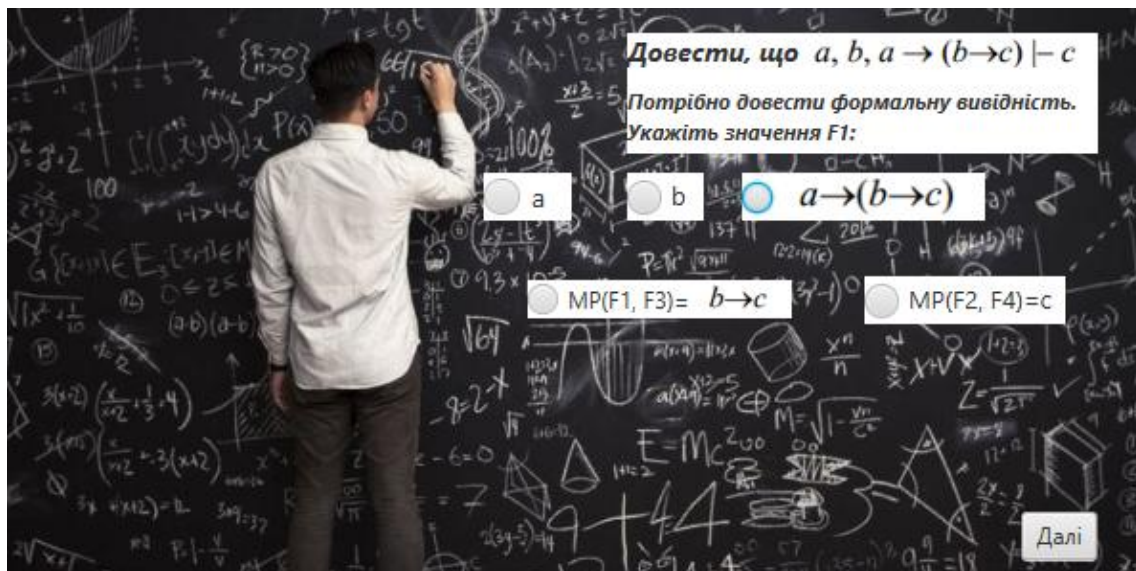


Рис. 4.29 – Друге практичне завдання

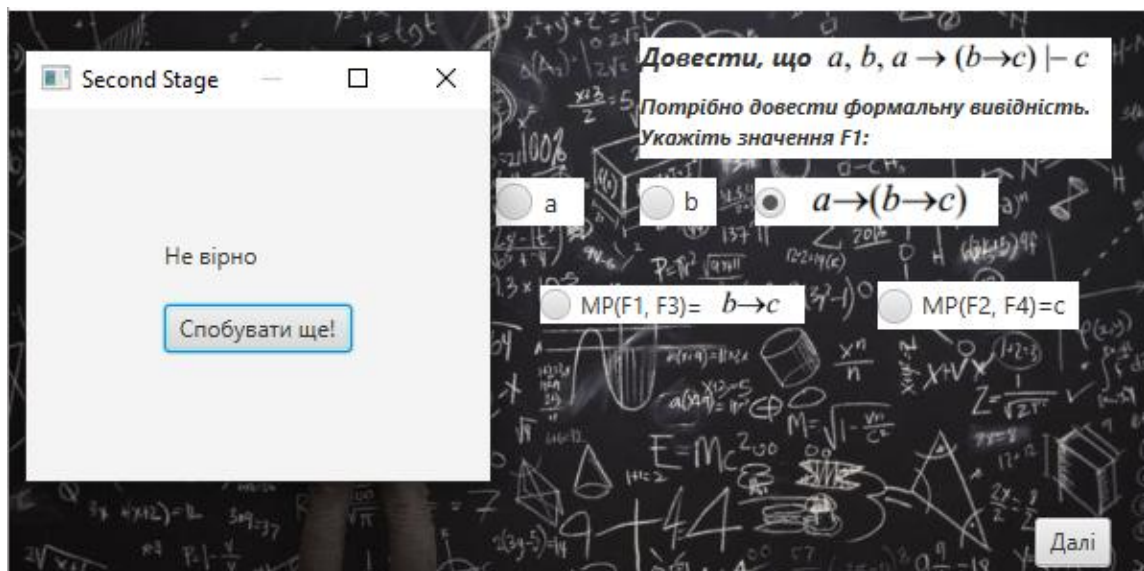


Рис. 4.30 – Повідомлення про помилку

Далі користувачу потрібно провести розбиття дужок на пари і вказати область дії кожного оператора у формулі $((A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C))))$ (рис.2.31).

Оскільки розбиття дужок на пари здійснюється індексами дужок, то користувачеві із запропонованої формули потрібно підставити правильно індекси.

Якщо користувач не підставить усі значення то висвітиться вікно «Не вірно», та запропонує користувачеві спробувати ще раз (рис.2.32).

Приклад : $(_1(_2A \rightarrow B_2) \rightarrow (_3(_4A \rightarrow C_4) \rightarrow (_5A \rightarrow (_6B \wedge C_6)_5)_3)_1)$.

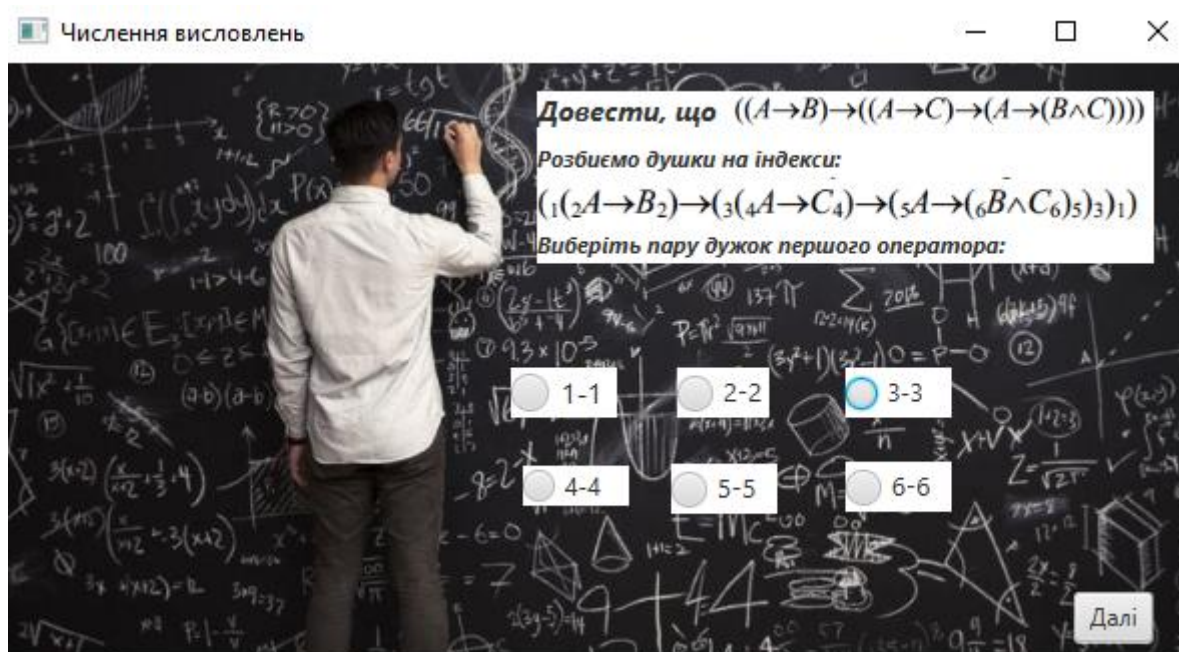


Рис. 4.31 – Третє практичне завдання

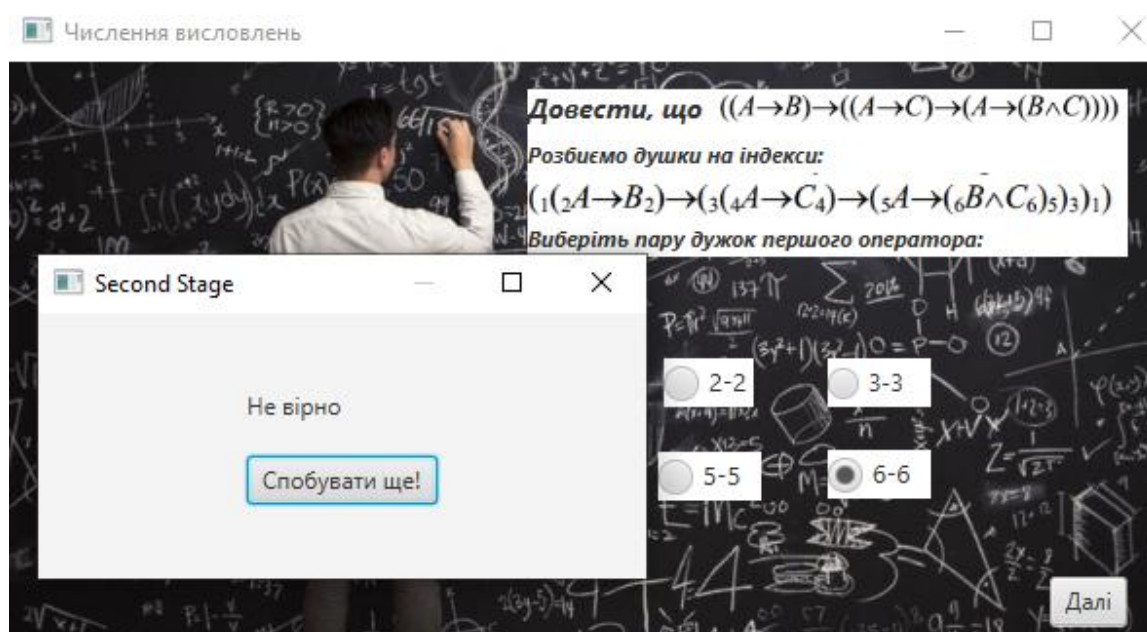


Рис. 4.32 – Повідомлення про помилку

4.4 Необхідна користувачу програми інструкція

Для початку роботи з програмою необхідно щоб на комп'ютері були встановлені бібліотеки java. Скачати актуальну версію можна с сайту java.com. Потім можна запустити файл тренажер з теми «Числення висловлювань.jar». Після запуску програми з'явиться меню вибору мови програми, де потрібно

обрати, на якій мові буде інтерфейс тренажеру (рис. 4.33). Після запуску з'явиться головне вікно програми з кнопкою «Розпочати» за допомогою якої почнуться тести (рис. 4.34).



Рисунок 4.33 – Вибір мови тренажера



Рис. 4.34 – Головне вікно програми

Після натискання кнопки «Розпочати» почнеться тестування (рис.4.35).

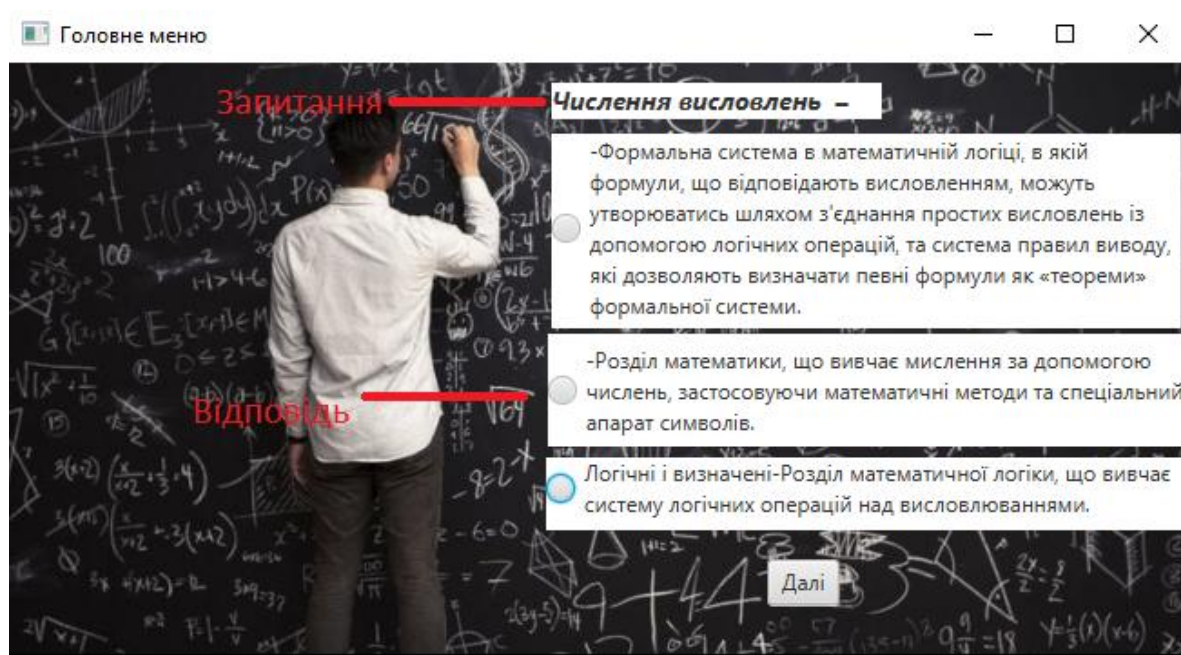


Рис. 4.35 – Вигляд тестової частини

4.5 Висновки за розділом 4

У результаті виконання практичної частини роботи можна зробити такі висновки:

- у даному розділі здійснено опис програмної реалізації;
- здійснено повне тестування програми;
- описано інструкції щодо користування з програмою.

ВИСНОВКИ

За результатами виконання магістерської роботи можна зробити такі висновки:

1. Закріплено лекційний матеріал по темі дистанційного навчання. Виконані поставлені задачі щодо оформлення та виконання магістерської роботи. Виконана поставлена задача по створенню тренажера з теми «Числення висловлювань».

2. Проведено аналіз недоліків и переваг дистанційного навчання. Оглянуто сучасне програмне забезпечення для створення тестів. Оглянуто вже існуючі програмні розробки.

3. Розглянуто теоретичні відомості по темі «Числення висловлювань». Поставлено практичні завдання та розроблений алгоритм програми. Обгрунтовано вибір програмної мови.

4. Виконано розбір розроблення програмного. Повністю проведено тестування програми по всім запитанням. Написано інструкцію по користуванні програмою.

5. Тренажер було впроваджено до системи дистанційної системи навчання ПУЕТ, що підтверджує акт про впровадження.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Дистанційне навчання [Електронний ресурс]: <https://uk.wikipedia.org/wiki>.
2. Ємець О. О. Методичні рекомендації щодо оформлення пояснювальних записок до дипломних проектів (робіт) для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» / О. О. Ємець, – Полтава : РВВ ПУЕТ, 2018. – 35 с.
3. Переваги та недоліки дистанційного навчання [Електронний ресурс]: <http://kerivnyk.info/perevahy-ta-nedoliky-dystantsijnoho-navchannya>.
4. Числення висловлювань [Електронний ресурс]: <https://uk.wikipedia.org/wiki>.
5. Числення висловлювань і предикатів [Електронний ресурс]: <http://www.kievoi.ippo.kubg.edu.ua/kievoi/logset/prepos.html>.
6. Числення висловлювань [Електронний ресурс]: <https://studfile.net/preview/3760003/>.
7. Про теореми числення висловлювань [Електронний ресурс]: <http://www.hups.mil.gov.ua/periodic-app/article/17120/ukr>.
8. Демида Б., Сагайдак С., Копил І. Системи дистанційного навчання: огляд, аналіз, вибір // Вісник Національного університету "Львівська політехніка". Комп'ютерні науки та інформаційні технології. – 2011. – № 694. – С. 98–107.
9. Кухаренко В.М., Рибалко О.В., Сиротенко Н.Г. Дистанційне навчання. Умови застосування. Дистанційний курс. За ред. Кухаренко В.М. – Харів: Торсінг, 2001. – 320 с.
10. Теория и практика дистанционного обучения: Учеб. пособие / Под ред. Е.С. Полат. - М.: Изд. Центр "Академия", 2004.
11. Информационные технологии и средства дистанционного обучения: Учеб. пособие для студентов высших учебных заведений / Ильдар Маратович Ибрагимов; Под ред. А.Н. Ковшова. - М.: Издательский центр "Академия", 2005.
12. Андреев А.А Компьютерные и телекоммуникационные технологии в сфере образования // Школьные технологии, 2001. №3.

13. Биков В. Ю. Дистанційне навчання в країнах Європи та США і перспективи для України // Інформаційне забезпечення навчально-виховного процесу: інноваційні засоби технології : монографія / [В. Ю. Биков, О. О. Гриценчук, Ю. О. Жук та ін.] / Академія педагогічних наук України, Інститут засобів навчання. – Київ : Атіка, 2005. – С. 77–140.
14. Клокар Н. Методологічні основи запровадження дистанційного навчання в системі підвищення кваліфікації / Н. Клокар // Шлях освіти. – 2007. – №4 (46). – С. 38–41.

ДОДАТОК А

Програмний код

Main.java

```
package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root = FXMLLoader.load(getClass().getResource("ChooseLang.fxml"));
        primaryStage.setTitle("Головне меню/Main Menu");
        primaryStage.setScene(new Scene(root, 500, 300));
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

GolovneMenu.fxml

```
package sample;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

public class Controller {

    @FXML
    private ResourceBundle resources;

    @FXML
    private URL location;

    @FXML
    private Button Button;

    @FXML
    void initialize() {
        assert Button != null : "fx:id=\"Button\" was not injected: check your FXML file 'GoloveMenu.fxml'.";
    }
}
```



```

        Button.setOnAction(event-> {
            Button.getScene().getWindow().hide();

            FXMLLoader loader = new FXMLLoader();
            loader.setLocation(getClass().getResource("1_vopros.fxml"));

            try {
                loader.load();
            } catch (IOException e) {
                e.printStackTrace();
            }

            Parent root = loader.getRoot();
            Stage Practic = new Stage();
            Practic.setTitle("Головне меню");
            Practic.setScene(new Scene(root));
            Practic.show();
        });
    }
}

```

Controller.java

```

package sample;

import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.geometry.Orientation;
import javafx.geometry.Pos;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.layout.FlowPane;
import javafx.scene.layout.StackPane;
import javafx.stage.Modality;
import javafx.stage.Stage;
import javafx.fxml.FXMLLoader;

public class Controller1 {

    @FXML
    private ResourceBundle resources;

    @FXML
    private URL location;

    @FXML
    private Button Button;

    @FXML
    private RadioButton RadioButton1;

    @FXML
    private RadioButton RadioButton2;

```

```

@FXML
private RadioButton RadioButton3;

@FXML
void initialize() {
    assert Button != null : "fx:id=\"Button\" was not injected: check your
FXML file '1_vopros.fxml'.";
    assert RadioButton1 != null : "fx:id=\"RadioButton1\" was not injected:
check your FXML file '1_vopros.fxml'.";
    assert RadioButton2 != null : "fx:id=\"RadioButton2\" was not injected:
check your FXML file '1_vopros.fxml'.";
    assert RadioButton3 != null : "fx:id=\"RadioButton3\" was not injected:
check your FXML file '1_vopros.fxml'.";

    ToggleGroup group = new ToggleGroup();
    RadioButton1.setToggleGroup(group);
    RadioButton2.setToggleGroup(group);
    RadioButton3.setToggleGroup(group);

    Button.setOnAction(event ->{
        if (RadioButton1.isSelected()){
            {
                RadioButton1.getScene().getWindow().hide();
                FXMLLoader loader = new FXMLLoader();
                loader.setLocation(getClass().getResource("2.fxml"));

                try {
                    loader.load();
                } catch (IOException e) {
                    e.printStackTrace();
                }

                Parent root = loader.getRoot();
                Stage Practic = new Stage();
                Practic.setTitle("Числення висловлень");
                Practic.setScene(new Scene(root));
                Practic.show();
            }
        }
        else if (RadioButton2.isSelected()){
            Label lb = new Label("Не вірно!\n ");
            Button bt = new Button ("Спобувати ще!");
            bt.setOnAction(new EventHandler<ActionEvent>() {
                @Override
                public void handle(ActionEvent event) {
                    bt.getScene().getWindow().hide();
                }
            });
            FlowPane root = new FlowPane(Orientation.VERTICAL, lb, bt);
            root.setAlignment(Pos.CENTER);
            Scene secondScene = new Scene(root, 500, 500 );

            // New window (Stage)
            Stage newWindow = new Stage();
            newWindow.setTitle("Second Stage");
            newWindow.setScene(secondScene);

            // Specifies the modality for new window.
            newWindow.initModality(Modality.APPLICATION_MODAL);
            newWindow.show();
        }
        else if (RadioButton3.isSelected()){
            Label lb = new Label("Не вірно!\n ");
            Button bt = new Button ("Спобувати ще!");

```

```

        bt.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                bt.getScene().getWindow().hide();
            }
        });
        FlowPane root = new FlowPane(Orientation.VERTICAL, lb, bt);
        root.setAlignment(Pos.CENTER);
        Scene secondScene = new Scene(root, 500, 500);

        // New window (Stage)
        Stage newWindow = new Stage();
        newWindow.setTitle("Second Stage");
        newWindow.setScene(secondScene);

        // Specifies the modality for new window.
        newWindow.initModality(Modality.APPLICATION_MODAL);
        newWindow.show();
    }
    else {
        Label lb = new Label("Оберіть відповідь!\n \n ");
        Button bt = new Button ("Повернутись назад!");
        bt.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                bt.getScene().getWindow().hide();
            }
        });
        FlowPane root = new FlowPane(Orientation.VERTICAL, lb, bt);
        root.setAlignment(Pos.CENTER);
        Scene secondScene = new Scene(root, 500, 500);

        // New window (Stage)
        Stage newWindow = new Stage();
        newWindow.setTitle("Second Stage");
        newWindow.setScene(secondScene);

        // Specifies the modality for new window.
        newWindow.initModality(Modality.APPLICATION_MODAL);
        newWindow.show();
    }
}

});
}
}

```